

**SOLVING THE INTEGRATED INVENTORY SUPPLY CHAIN
PROBLEMS USING META-HEURISTIC METHODS**

SEYED MOHSEN MOUSAVI

**FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2017

**SOLVING THE INTEGRATED INVENTORY SUPPLY CHAIN
PROBLEMS USING META-HEURISTIC METHODS**

SEYED MOHSEN MOUSAVI

**THESIS SUBMITTED IN FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF
PHILOSOPHY**

**FACULTY OF ENGINEERING
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2017

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Seyed Mohsen Mousavi (I.C/Passport No: XXXXXXXXXX)

Registration/Matric No: KHA130008

Name of Degree: DOCTOR OF PHILOSOPHY

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Solving the Integrated Inventory Supply Chain Problems Using Meta-heuristic
Methods

Field of Study: Optimization (Engineering and Engineering Trades)

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this work;
- (2) This work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently, and the title of the work and its authorship have been acknowledged in this work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature : Seyed Mohsen Mousavi
subscribed and solemnly declared before,

Date:

Witness's Signature

Date:

Name:

Designation:

SOLVING THE INTEGRATED INVENTORY SUPPLY CHAIN

PROBLEMS USING META-HEURISTIC METHODS

ABSTRACT

Nowadays, managing supply chain networks are really important for all companies especially those producing seasonal products. The novelties of this work are as follows. First, a novel multi-objective multi-item seasonal inventory control model was developed for known-deterministic variable demands where shortages in combination of backorder and lost sale were considered. Moreover, all unit discounts (AUD) for a number of products and incremental quantity discount (IQD) for some other items were considered. The main novelty of the first model is to minimize the required storage space for the first time in the literature in addition to minimizing the total inventory cost. While the weights of both objectives are considered to be fuzzy numbers due to their uncertainty, the model was formulated into a fuzzy multi-objective decision making (FMODM) framework called Fuzzy Weighted Sum Method (FWSM) and was shown to be a mixed-integer binary nonlinear programming type. In order to solve the model, a multi-objective particle swarm optimization (MOPSO) approach was applied. The efficiency of the algorithm was compared to a multi-objective genetic algorithm (MOGA) as well. The second novelty of the work is an extension of the proposed model, where a novel model of the integrated seasonal inventory-supply chain distributor–retailer network for a multiple-product location allocation problem in a planning horizon consisting of multiple periods was formulated. The distance between the distributors and retailers were assumed to be Euclidean and Square Euclidean. The retailers purchase the products from the distributors under both AUD and IQD policies. Furthermore, the products were delivered in packets of known size of items where the model was extended for both cases of with and without shortages. Besides, the

distributors (vendors) stored the manufactured products in their own warehouses before delivering them to the retailers since the total warehouse spaces and the total available budget for purchasing the items from the distributors were limited. It was considered that the distributors manufacture the products under some production limitations. The aim of the problem was to find the optimal order quantities of the products purchased by the retailers from the distributors in different periods and determine the coordinates of the distributors' locations to minimize the total inventory-supply chain cost. In fact, finding out the optimal order quantities of items in each period and the optimal locations of distributors among retailers are the main novelty of the second model proposed for the first time in the literature. As the mixed integer nonlinear model of the problem was complicated to solve using exact methods, several meta-heuristic algorithms were employed in to optimize the models under investigation. A Modified Particle Swarm Optimization (MPSO) algorithm, a Genetic Algorithm (GA), a modified fruit fly optimization algorithm (MFOA) and a simulated annealing (SA) algorithm were used to find the optimal solution. A design of experiment approach i.e. Taguchi was used to optimize the algorithms parameters. While there was no benchmark in the literature, some numerical examples were generated to show the performance of the algorithms for both Euclidean and Square Euclidean distances while some case studies were also considered.

Keywords: inventory control problem, seasonal items, supply chain, meta-heuristics

ABSTRAK

Pada masa kini, pengurusan dan pengendalian rangkaian bekalan adalah penting bagi semua syarikat terutamanya yang menghasilkan produk bermusim dan berfesyen. Dalam kajian ini, model kawalan inventori bagi pelbagai-item pelbagai-tempoh produk bermusim dibangunkan untuk permintaan berubah deterministik-berketentuan di bawah bajet yang terhad. Selain itu, semua diskaun unit (AUD) untuk beberapa produk dan diskaun kuantiti tambahan (IQD) untuk beberapa barangan lain juga dipertimbangkan. Walaupun objektif adalah untuk mengurangkan jumlah kos inventori dan ruang simpanan yang diperlukan, model itu dirumuskan ke dalam rangka kerja kriteria berbilang membuat keputusan kabur (FMODEM) dan ditunjukkan sebagai jenis pengaturcaraan linear binari integer bercampur. Dalam usaha untuk menyelesaikan model, pendekatan pelbagai objektif pengoptimuman swarm zarah (MOPSO) digunakan. Satu set penyelesaian kompromi termasuk penyelesaian optimum dan yang hampir optimum menggunakan MOPSO telah diperolehi untuk beberapa contoh berangka, di mana hasilnya dibandingkan dengan keputusan yang diperolehi dengan menggunakan pendekatan kabur berpemberat. Untuk menilai kecekapan MOPSO yang dicadangkan, model ini juga diselesaikan menggunakan algoritma genetik pelbagai objektif (MOGA). Sebagai lanjutan daripada model yang dicadangkan, reka bentuk bersepadu rangkaian inventori- pengedar rantai bekalan-peruncit untuk masalah peruntukan lokasi dalam tempoh rancangan pembangunan yang terdiri daripada pelbagai-tempoh bagi pelbagai-item bermusim dimodelkan. Jarak antara pengedar dan peruncit diandaikan sebagai Euclidean dan Euclidean persegi. Peruncit membeli produk dari pengedar di bawah kedua-dua polisi AUD dan IQD. Di samping itu, produk yang dihantar dalam paket yang diketahui saiz produk di mana model ini diperluaskan untuk kedua-dua kes dengan dan tanpa kekurangan bekalan dimana

jika berlaku kekurangan, sebahagian daripada permintaan dianggap backorder dan sebahagian lagi dikira sebagai lost sales. Selain itu, pengedar menyimpan produk dalam gudang mereka sendiri sebelum diserahkan kepada peruncit kerana jumlah ruang gudang dan jumlah bajet yang ada untuk membeli barangan dari pengedar (vendor) adalah terhad. Kekangan kapasiti juga diambil kira apabila merancang tahap inventori. Adalah dianggap bahawa pengedar mengeluarkan produk di bawah beberapa batasan pengeluaran. Oleh kerana masalah model integer campuran tak linear rumit untuk diselesaikan menggunakan kaedah yang tepat, beberapa algoritma meta-heuristik diambil kira dalam penyelesaian ini untuk mengoptimumkan model yang dikaji. Dalam model tanpa kekurangan bekalan, algoritma Modified Particle Swarm Optimization (MPSO) dan algoritma genetik (GA) digunakan untuk menyelesaikan masalah itu. Berbanding kes kekurangan bekalan, pengoptimuman algoritma lalat buah diubah suai (MFOA) dicadangkan untuk mencari penyelesaian yang optimum. Dalam model ini, kerana sifat yang tak linear rumusan asal dan perasan bahawa tidak ada penanda aras disediakan dalam kesusasteraan untuk membenarkan dan mengesahkan keputusan, pengoptimuman swarm zarah (PSO) dan penyepuhlindapan simulasi (SA) algoritma diwakili juga. Dalam usaha untuk mengoptimumkan parameter algoritma yang dicadangkan itu, reka bentuk eksperimen iaitu pendekatan Taguchi digunakan. Beberapa contoh berangka dijana untuk menunjukkan prestasi dan penggunaan algoritma untuk kedua-dua Euclidean dan jarak Euclidean persegi. Dalam model ini, kerana sifat yang tak linear pada rumusan asal dan pemerhatian terhadap kekurangan penanda aras disediakan dalam kajian lepas untuk membenarkan dan mengesahkan keputusan, algoritma pengoptimuman swarm zarah (PSO) dan penyepuhlindapan simulasi (SA) juga disediakan. Beberapa contoh berangka dijana untuk

menunjukkan prestasi dan penggunaan algoritma untuk kedua-dua jarak Euclidean dan Euclidean persegi.

Katakunci: model kawalan inventori, produk bermusim, rangkaian bekalan, meta-heuristics

ACKNOWLEDGEMENTS

All praise and thanks to our Creator, and Sustainer God, who is always most beneficent and most gracious.

I would like to take this opportunity to express my deepest appreciation to my supervisor, Dr. Siti Nurmaya and to my external supervisor Dr. Ezutah Udony Olugu for their support, valuable suggestions and for encouraging me to keep going. I appreciate their open mindedness and vast knowledge, which they always made available to me. They have been a very generous source of knowledge and support, and a role model to follow. I am indebted to them forever. I would also like to express deep gratitude to my old supervisors Professor Dr. Ardeshir Bahreininejad, for all his benign guidance and invaluable support during this research, and for having faith in me to be able to achieve this accomplishment.

Also, I appreciate University of Malaya (UM), Kuala Lumpur, Malaysia, for supporting me financially through Bright Sparks research grant. In addition, I would like to thank all members in the Department of Mechanical Engineering who supported me directly or otherwise.

Enough appropriative words could not be found to express sincere appreciation to my wife for her endless patience, understanding, friendliness, encouragement and absolute love in all difficulties in research and living. I dedicate this thesis to her.

With best wishes to all of them,

Seyed Mohsen Mousavi

Author

TABLE OF CONTENTS

Abstract.....iii

Abstrak.....v

Acknowledgements viii

Table of Contents ix

List of Figures.....xv

List of Tables.....xix

List of Symbols and Abbreviations..... xxiii

List of Appendices xxv

CHAPTER 1: INTRODUCTION..... 1

1.1 Research Background..... 1

1.2 Statement of the Problem 2

1.3 Research Objectives and Research Questions..... 7

1.4 Scope of the Research 8

1.5 Organization of the Thesis 9

CHAPTER 2: LITERATURE REVIEW..... 11

2.1 Introduction 11

2.2 Inventory Control Problem (ICP)..... 11

2.3 Inventory-Supply chain Problems (ISP) 11

2.3.1 Multi-objective ISP 11

2.3.1.1 Multi-product ISP..... 12

2.3.1.2 Multi-period ISP..... 15

2.3.1.3	Discounted ISP	18
2.3.1.4	Fuzzy ISP	20
2.3.2	Single-objective ISP	25
2.3.2.1	Inventory-Supply Chain Management Location Allocation (ISLA)	26
2.3.2.2	Location-Allocation Problem (LAP)	29
2.3.2.3	Inventory-Supply Chain Location-Allocation without shortages (ISLAWOS)	30
2.3.2.4	Inventory-Supply Chain Location-Allocation with shortages (ISLAWS)	33
2.3.3	Lot-Sizing ISLA (LSISLA)	37
2.4	Solution Methods for Solving ISLA and LSISLA	40
2.4.1	PSO and GA	41
2.4.2	FOA	45
2.4.3	Taguchi approach.	47
2.5	Summary	49
CHAPTER 3: METHODOLOGY		53
3.1	Introduction	53
3.2	Research Steps	53
3.3	The proposed multi-item multi-period ICP	56
3.4	The proposed multi-item multi-period ISLAWOS	58
3.5	The proposed multi-item multi-period ISLAWS	59
3.6	Solution methods	60

3.6.1	Particle Swarm Optimization	63
3.6.2	Genetic Algorithm.....	64
3.6.3	Fruit Fly Optimization.....	66
3.6.4	Simulated Annealing	67
3.6.5	Taguchi method.....	68
3.7	Summary	69
 CHAPTER 4: MULTI-ITEM MULTI-PERIODIC INVENTORY CONTROL		
PROBLEM WITH VARIABLE DEMAND AND DISCOUNTS: A		
PARTICLE SWARM OPTIMIZATION ALGORITHM.....71		
4.1	Introduction	71
4.2	Problem definition, assumptions, and notations	72
4.3	Problem formulation	79
4.3.1	The objective functions	79
4.3.2	The constraints	84
4.4	The proposed MOPSO	87
4.4.1	Generating and initializing the particles positions and velocities	87
4.4.2	Selecting the best position and velocity	88
4.4.3	Velocity and position update.....	89
4.4.4	Stopping criterion.....	90
4.5	The proposed MOGA.....	91
4.6	Numerical illustrations	95
4.6.1	The generated data.....	96
4.6.2	A Case Study.....	109

4.7	Summary	114
-----	---------------	-----

CHAPTER 5: A MODIFIED PARTICLE SWARM OPTIMIZATION FOR SOLVING THE INTEGRATED LOCATION AND INVENTORY CONTROL PROBLEMS IN A TWO-ECHELON SUPPLY CHAIN NETWORK 116

5.1	Introduction	116
5.2	Problem description and formulation.....	117
5.2.1	Notations and assumptions.....	119
5.2.2	Problem formulation	122
5.3	Solution methodologies.....	125
5.3.1	Initializing the parameters.....	125
5.3.2	Evaluating the particles	126
5.3.3	Updating the velocities and positions	126
5.3.4	Stopping criteria	127
5.4	Data generating, Parameter setting and computational results	129
5.4.1	Generating random data	129
5.4.2	Parameter setting	131
5.4.3	The results and comparisions	134
5.4.4	A Case Study.....	141
5.5	Summary	147

CHAPTER 6: OPTIMIZING A LOCATION ALLOCATION INVENTORY PROBLEM IN A TWO-ECHELON SUPPLY CHAIN

NETWORK: A MODIFIED FRUIT FLY OPTIMIZATION	
ALGORITHM.....	149
6.1 Introduction	149
6.2 Modeling the proposed problem	151
6.2.1 Indices, parameters, variables, and decision variables.....	152
6.2.2 The supply chain cost.....	155
6.2.3 The constraints	157
6.3 Solving methodologies.....	160
6.3.1 Initializing and representing the flies	161
6.3.2 Smell-based and vision-based search.....	162
6.3.3 Global vision-based search	162
6.3.4 Stopping criterion.....	163
6.4 Testing and comparisons	163
6.4.1 The generated data	163
6.4.2 A Case Study.....	181
6.5 Summary	186
CHAPTER 7: CONCLUSION AND RECOMMENDATIONS	187
7.1 Conclusion.....	187
7.2 Contributions.....	189
7.3 Implementations to the study	190
7.4 Recommendations for Future Research	191
7.5 Shortcomings and Strengths.....	192
References.....	193

LIST OF PUBLICATIONS AND PAPERS PRESENTED.....213

Appendix.....215

LIST OF FIGURES

Figure 1.1: A representation of the supply chain network	2
Figure 1.2: Graphical representation of the first problem.....	6
Figure 1.3: Graphical representation of the second problem	6
Figure 1.4: Graphical representation of the third problem.....	6
Figure 3.1: Methodological flowchart of this research	56
Figure 3.2: The flowchart of the proposed PSO algorithm	64
Figure 3.3: The flowchart of the proposed GA	66
Figure 3.4: The flow chart of the proposed MFOA	67
Figure 3.5: The pseudo-code of the proposed SA	68
Figure 4.1: A graphical illustration of the proposed inventory control problem ...	76
Figure 4.2: Some possible situations for the inventory of item i in 5 periods	79
Figure 4.3: AUD policy for purchasing the products in different periods	82
Figure 4.4: IQD policy for purchasing the products in different periods.....	83
Figure 4.5: The pseudo code of MOPSO algorithm	91
Figure 4.6: The structure of a chromosome	93
Figure 4.7: The uniform crossover operation.....	94
Figure 4.8: The one-point mutation operation	95
Figure 4.9: The structure of a particle	95

Figure 4.10: The triangular fuzzy numbers	102
Figure 4.11: The α -cut ($\alpha = 0.5$) procedure applied for the proposed problems with triangular fuzzy numbers.....	103
Figure 4.12: The mean S/N ratio plot for parameter levels of MOPSO and MOGA in Problem 7 of Table 4.2.....	105
Figure 4.13: The pictorial representation of the performances of the algorithms...	106
Figure 4.14: The Box-plot and the individual value plot of the performances of the algorithms	107
Figure 4.15: The residual plots of the algorithms	108
Figure 4.16: The ANOVA analysis of the CPU time for both the algorithms.....	108
Figure 5.1: The illustration of the supply chain model	119
Figure 5.2: Some possible scenarios for the available inventory system.....	123
Figure 5.3: The presentation of a particle	126
Figure 5.4: The representation of a population of the articles	126
Figure 5.5: The mean S/N ratio plot of the MPSO on Problem No. 1 of Small-scale	133
Figure 5.6: The mean S/N ratio plot of the GA on Problem No. 1 of Small-scale	133
Figure 5.7: The graphical representation of the objective function resulted from MPSO and GA on the generated instances of (a) Small-scale, (b) Medium-scale and (c) Large-scale	139

Figure 5.8: A representation of the optimal solution obtained by MPSO for Problem No. 1 of Small-scale category	140
Figure 5.9: The convergence path of MPSO for Problem No. 10 of Medium-scale category	140
Figure 5.10: The convergence path of GA for Problem No. 10 of Medium-scale category.....	141
Figure 5.11: The optimal location of the vendors among the buyers obtained by MPSO for problem No. 7 of Medium-scale category	141
Figure 5.12: The coordinates of the buyers' locations proposed by the marketing department of ORAMAN Wood Industry shown on Google Maps.....	143
Figure 5.13: The optimal location of the vendors among the buyers obtained by MPSO for the case study.....	147
Figure 5.14: The optimal location of the vendors among the buyers obtained by GA for the case study	147
Figure 6.1: The proposed supply chain network	151
Figure 6.2: The graphical illustration of the inventory system in the proposed supply chain network	151
Figure 6.3: The representation of a fruit fly	161
Figure 6.4: The representation of a population of the flies	161
Figure 6.5 : The mean S/N ratio plot for different levels of the parameters for <i>Prob. No. 11</i> of medium-size for MFOA, PSO, and SA algorithms with Euclidean distance.....	172

Figure 6.6: The mean S/N ratio plot for different levels of the parameters for <i>Prob.</i> <i>No. 11</i> of medium-size for MFOA, PSO, and SA with square Euclidean distance.....	173
Figure 6.7: The graphical illustration of the fitness values of the algorithms with Euclidean distance for (a1) small-size, (b1) medium-size, and (c1) large- size problems.....	174
Figure 6.8: The graphical illustration of the fitness values of the algorithms with square Euclidean distance for (a2) small-size, (b2) medium-size, and (c2) large-size problems	175
Figure 6.9: The box-plot of the proposed algorithms for (S1) small-size, (M1) medium-size, and (L1) large-size problems with Euclidean distances	176
Figure 6.10: The box-plot of the proposed algorithms for (S2) small-size, (M2) medium-size, and (L2) large-size problems with square Euclidean.....	177
Figure 6.11 : The optimal location of the producers among the retailers obtained by MFOA for the case study.....	183
Figure :6.12 The optimal location of the producers among the retailers obtained by PSO for the case study.....	184
Figure 6.13: The optimal location of the producers among the retailers obtained by SA for the case study.....	185

LIST OF TABLES

Table 2.1: Literature review of some of the works related to Section 2.3	23
Table 2.2: A brief review of the works related to ISLAWS in the literature	51
Table 4.1: The input data of the test problems	97
Table 4.2: Different problems and their optimal TMF values obtained by the two algorithms.....	98
Table 4.3: The general data for a problem with 5 items and 3 periods.....	100
Table 4.4: The parameters of the two algorithms and their levels	100
Table 4.5: The Taguchi L_9 design along with objective values of the algorithm...104	
Table 4.6: The optimal levels of the algorithms' parameters for Problem 7 of Table 4.2.....	104
Table 4.7: The best result of the MOPSO algorithm	106
Table 4.8: The best result of the MOGA algorithm	106
Table 4.9: The ANOVA analysis of the performances	106
Table 4.10: The collected data for the case study with 4 items and 4 periods.....	111
Table 4.11: The optimal solutions obtained by MOGA for the case study.....	115
Table 4.12: The optimal solutions obtained by MOPSO for the case study	115
Table 5.1: The input data of the test problems	130
Table 5.2: Sizes of the proposed instances.....	130
Table 5.3: The MPSO and GA parameters' levels.....	132

Table 5.4: The experimental results on the MPSO and GA parameters for Problem No. 1 of Small-scale	133
Table 5.5: The input parameters and the objective function of MPSO and GA for Small-scale problems.....	135
Table 5.6: The input parameters and the objective function of MPSO and GA for Medium-scale problems	136
Table 5.7: The input parameters and the objective function of MPSO and GA for Large-scale problems.....	137
Table 5.8: The ANOVA for the instances of Small-scale category	140
Table 5.9: The ANOVA for the instances of Medium-scale category.....	140
Table 5.10: The ANOVA for the instances of Large-scale category	140
Table 5.11: The data for a real problem collected from the first buyer	144
Table 5.12: The data for a real problem collected from the second buyer.....	144
Table 5.13: The data for a real problem collected from the third buyer	144
Table 6.1: General data for small-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances.....	167
Table 6.2: Some general data for medium-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances.....	168

Table 6.3: Some general data for large-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances.....	169
Table 6.4: The parameter levels of the three algorithms for <i>Prob. No. 11</i> of medium-size with Euclidean distance	170
Table 6.5: The parameters levels of the three algorithms for <i>Prob. No. 11</i> of medium-size with square Euclidean distance	170
Table 6.6: The optimal levels of the parameters of the three algorithms for <i>Prob. No. 11</i> of medium-size with Euclidean distance	171
Table 6.7: The optimal levels of the parameters of the three algorithms for <i>Prob. No. 11</i> of medium-size with square Euclidean distance.....	171
Table 6.8: The one-way ANOVA to compare the algorithms for small-size problems with Euclidean distance	178
Table 6.9: The one-way ANOVA to compare the algorithms for small-size problems with square Euclidean distance.....	178
Table 6.10: The one-way ANOVA to compare the algorithms for medium-size problems with Euclidean distance	178
Table 6.11: The one-way ANOVA to compare the algorithms for medium-size problems with square Euclidean distance	178
Table 6.12: The one-way ANOVA to compare the algorithms for large-size problems with Euclidean distance	179
Table 6.13: The one-way ANOVA to compare the algorithms for large-size problems with square Euclidean distance	179

Table 6.14: CPU times (s) taken by each algorithm to solve small-size problems	
.....	179
Table 6.15: CPU times (s) taken by each algorithm to solve medium-size problems	
.....	180
Table 6.16: CPU times (s) taken by each algorithm to solve large-size problems	
.....	180
Table 6.17: The backordering and lost sale costs of the case study explained in section 5.4.3.....	181

LIST OF SYMBOLS AND ABBREVIATIONS

EOQ	:	Economic order quantity
PSO	:	Particle Swarm Optimization
AUD	:	All unit discount
IQD	:	Incremental quantity discount
ISLA	:	Inventory-supply chain location allocation
FMODM	:	Fuzzy multi-objective decision making
FWSM	:	Fuzzy Weighted Sum Method
FOA	:	Fruit fly optimization algorithm
MFOA	:	Modified fruit fly optimization algorithm
MOGA	:	Multi-objective genetic algorithm
MOPSO	:	Multi-objective particle swarm optimization
GA	:	Genetic Algorithm
SA	:	Simulated Annealing
ICP	:	Inventory control problem
ACO	:	Ant colony optimization
SCM	:	Supply chain management
EDI	:	Electronic Data Interchanges
NSGA II	:	Non-dominated sorting genetic algorithm
LAP	:	Location allocation problem
LA	:	Location allocation
ISLAWOS	:	Inventory-supply chain location allocation without shortages
MPSO	:	Modified particle swarm optimization
ISLAWS	:	Inventory-supply chain location allocation with shortages
ISP	:	Inventory-Supply chain problem

LSISLA	:	Lot-Sizing Inventory-Supply chain location allocation problem
N	:	No
Y	:	Yes
S/N	:	Signal/Noise
TB	:	Total Budget

LIST OF APPENDICES

APPENDIX A: The data sheets collected from the company.....216

APPENDIX B: The data sheet collected from the company relevant to the buyers
.....218

APPENDIX C: The MATLAB codes of the algorithms proposed for solving the
problems.....225

CHAPTER 1: INTRODUCTION

1.1 Research Background

Planning for inventory order quantity and the order time of the products is an important managerial decision that affects the total costs associated with an inventory system. The most basic model that has been developed for this decision making so far is the economic order quantity (EOQ). While there is not a common agreement on the history of EOQ, (Roach, 2005) refers the history of EOQ back to Ford W. Harris who has developed it in 1913. This model in its classic form encompasses the planning for one product in a period with many assumptions. However, while these assumptions make the model simpler, they limit its applicability in real-world environments. To overcome this limitation, researchers infract the traditional assumptions gradually.

In order to extend the classical EOQ, one approach is to consider the inventory problem of multiple products, multiple planning periods, or their combination. In this regard, (Lee & Kang, 2008) developed a model to manage inventory of a product in multiple periods. Roy and Maiti (2000) proposed a multi-item inventory model with constant demand and infinite replenishment under the restrictions on storage area, total average shortage cost, and total average inventory investment cost. Silver and Moon (2001) developed a constrained optimization model for a group of end items with known and constant demand rate along with convertibility to other useful units. Panda et al. (2005) proposed a nonlinear goal programming technique to obtain the EOQ of a multi-item inventory problem using penalty functions in a decision-making context. Kim and Kim (2000) formulated a multi-period inventory/distribution planning problem as a mixed integer linear

programming and solved it by a Lagrangian relaxation approach. Luciano et. al. (2003) used the value-at risk idea for inventory management and provided a risk measure for inventory management in a static multi-period framework. Matsuyama (2006) tried to generalize the newsboy model to deal with unsatisfied demands or unsold quantities; extending the planning horizon to more than one period.

Supply chain is an integrated system of facilities and activities that synchronizes inter-related business functions of material procurement, material transformation to intermediates and final products, and distribution of these products to customers (Simchi-Levi et. al., 2000). Supply chain management is a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system-wide costs while satisfying service level requirements (Simchi-Levi et al., 2000). Figure 1.1 shows a representation of the supply chain network proposed in this work in which suppliers, manufacturers, distributors, retailers and customers are included.

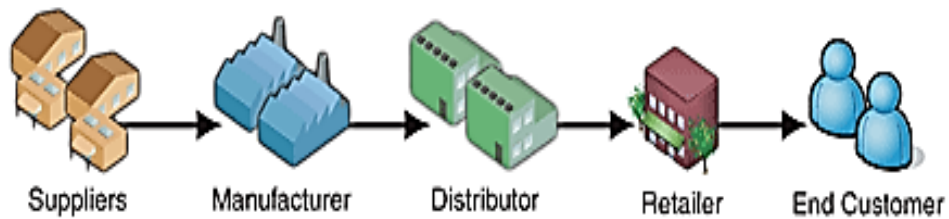


Figure 1.1: A representation of the supply chain network

1.2 Statement of the Problem

Designing an appropriate supply chain network and connections among the members of the network can play an important role in reducing the overall costs of production. It also should be taken into account that producing a suitable level of the products based on the orders received from customers can increasingly reduce

the costs in a supply chain. Since the marketing teams assess the location of the customers in different areas to locate the distributors, distance can be a determining factor to reduce the transportation costs with respect to the volume of the products transferred from distributors to retailers. Therefore, determining the proper location of the distributors in between retailers to transfer the products ordered by the retailers with the minimum cost and in the shortest time are two vital decisions that the managers should take into consideration in order to effectively manage the supply chains. Investigating these kinds of decisions, without taking into account certain tactical and operational decisions, could have a negative impact on the performance of the supply chain. In a supply chain network, making decisions on enterprise matters straightly affects transportation and costs involved in inventory system. Therefore, optimizing an integrated supply chain model with more flexibility and efficiency is required. In the particular case of the products, those are manufactured and stored during different periods (seasons), some key issues give the relevance of this problem. On the one hand, these kinds of products will be *démodé* (for clothes) or spoiled (i.e. chemical or food materials) after a specific time-period. It means that these products are only popular in those particular seasons (periods). On the other hand, transporting and holding the products in different periods need to be paid more attention in comparison with other items due to expiration date among other reasons.

The growth and developments in logistics and supply chain caused a more attention to controlling and managing inventory systems. This is evidence in the attention they have received from many researchers and practitioners. These attentions appear as efforts on inventory management to run various strategies of supply chain problem. One of the most representative strategies of supply chain problems is the well-known facility location–allocation problem which is

formulated to be considered as an inventory-supply chain location allocation (ISLA) model.

In recent decades, scientists have been mimicking natural phenomena to propose methods and algorithms for solving complex optimization problems. Based on the complexity of real-life optimization problems, one may not be able to use exact algorithms. Typically, meta-heuristic methods are frequently used to find a near optimum solution in an acceptable period.

Meta-heuristics are kind of near-optimal algorithms that were proposed in the recent years to integrate basic heuristic methods in higher-level structures in order to effectively and efficiently search a solution space. Nowadays, these algorithms have a large number of applications in optimization of different hard-to-solve problems. The particle swarm optimization (PSO) proposed by (Eberhart & Kennedy, 1995) is a population-based stochastic meta-heuristic algorithm that was inspired by social behavior of bird flocking or fish schooling. PSO is a meta-heuristic that requires few or no assumptions on the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as PSO do not guarantee an optimal solution is ever found. PSO can therefore be used on optimization problems that are partially irregular and noisy over time (Gigras & Gupta, 2012). In the past decade, PSO has been well applied to solve different problems. In inventory control problems, (Taleizadeh et. al., 2010) solved an integer nonlinear programming type of inventory control problem using PSO. Baykasoglu & Gocken (2011) utilized a PSO to solve an EOQ inventory control problem in fuzzy environments.

The Fruit Fly Optimization Algorithm (FOA) is a novel meta-heuristic algorithm for finding out global optimization based on the food finding behavior of the fruit

fly. The fruit fly itself is superior to other species in sensing and perception, especially in osphresis and vision (Pan, 2012).

In this research, the main focus would be on the modeling and formulating novel inventory and supply chain problems for multiple items in a finite horizon (multi-period) considering the limitations inherent in the real world problems. The proposed model can be also useful for seasonal and fashion industry problems those manufacturing several items in multi-period (season). In fact, multi-period or finite horizon means that the process of ordering (replenishment) items starts in a particular period and will finish in another specific period. The ordering process will be placed for each item in each period according to the demand of the customers received for that particular item in each period. This process can be different for different type of items. For example in fashion clothing products, the ordering (replenishment) process of the warm clothes as well as jackets, cardigan, gloves, caps and mufflers will start in the early of December and finish by the end of February in Tehran (Iran). The demands of these items are variable and do not follow a pattern trend at all. In this research work, first a mixed-integer binary nonlinear programming is formulated to model a multi-product multi-period inventory control problem where both the total inventory costs and the required storage are minimized. Two discount policies which are which are all-unit discount (AUD) for a number of products and incremental quantity discount (IQD) for some other items are considered by sellers. Then, a location allocation problem under a supply chain network is modeled since the aim is to find the location of the vendors and the order quantities of the products (allocation) sold to the retailers. In the proposed inventory-supply chain problem, after producing the products, the distributors store the products into their warehouses and then sell them to the

retailers. The graphical representation of the proposed first, second and third problems are shown by Figures 1.2, 1.3 and 1.4, respectively.

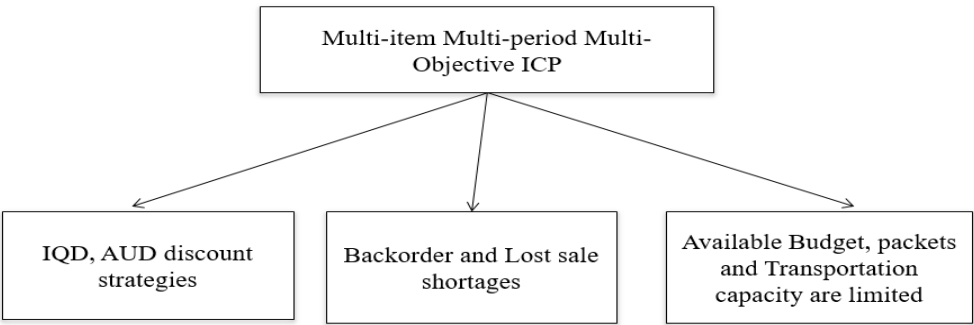


Figure 1.2: Graphical representation of the first problem

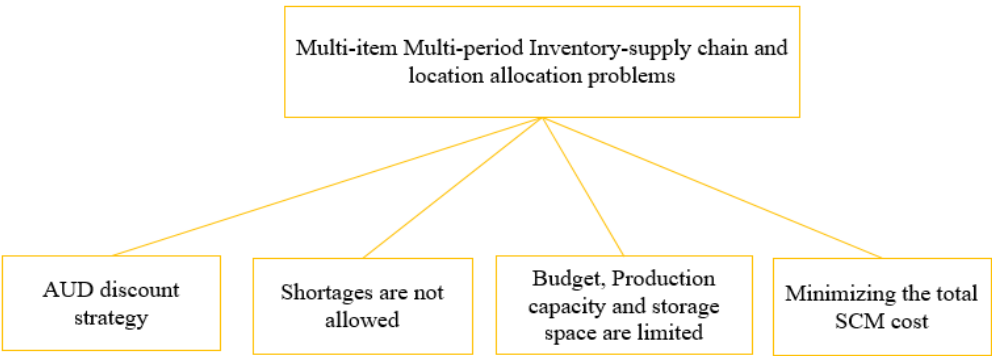


Figure 1.3: Graphical representation of the second problem

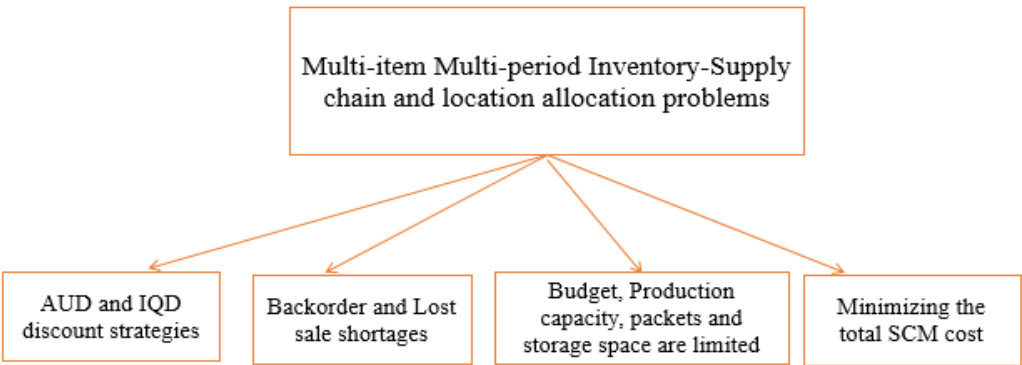


Figure 1.4: Graphical representation of the third problem

Figure 1.2 shows a graphical representation of the first problem where a multi-objective fuzzy multi-item multi-period inventory control problem with its constraints is presented in details. A seasonal multi-item inventory-location

allocation problem for a two-echelon supply chain is shown in Figure 1.3 where shortages are not allowed. Figure 1.4 depicts the graphical presentation when shortages are allowed where the problem has its own limitations.

The research problems of the proposed thesis are listed as follows:

- There is no any fuzzy multi-objective model for a multi-period (seasonal) inventory problem in the literature in which the objectives are minimizing total costs and the required storage space for a new storage, simultaneously.
- Lack of a multi-period (seasonal) inventory model for locating some new producers among the retailers in a two-echelon supply chain problem.
- Finding the most suitable solution methodologies for solving these problems.

Here, seasonality means those items which their ordering (replenishment) process will start in a particular period and will finish in another specific period based on the demands received (Dayarian et al., 2016; Diewert et al., 2009; Mogale et al., 2017; Saracoglu et al., 2014; Tanksale & Jha, 2017). In other words, in this work, the definition of seasonality is interchangeable to multi-period items referred to (Dayarian, et. al., 2015) definition “A convenient way to model the seasonal fluctuations is to represent the horizon as a finite set of periods”.

1.3 Research Objectives and Research Questions

The objectives of this research are listed as follows:

- i. To formulate mathematical models for a fuzzy multi-objective multi-product multi-period (seasonal) inventory control problem.
- ii. To formulate a mathematical model for a two-echelon multi-product multi-period (season) inventory-supply chain location allocation problem.

- iii. To use optimization approaches in order to reach the optimum or near-optimum solutions by employing well-known meta-heuristic algorithms and compare their performances.

The research questions of the study are listed as follows:

- I. How a fuzzy multi-objective multi-item multi-period (seasonal) inventory problem can be modeled.
- II. How a two-echelon multi-product multi-period (seasonal) inventory-supply chain location allocation problem can be modeled.
- III. What are the solutions to the proposed inventory and supply chain problems.

1.4 Scope of the Research

The novelty of the research was developed in two directions. First part focused on a fuzzy multi-objective multi-product multi-period inventory control problem with variable demands in which the total available budget for purchasing the products and also the truck capacity to transfer the products were restricted. While the objectives were to minimize both the total inventory cost and the required storage space, the model was formulated into a fuzzy multi-objective decision making (FMODM) framework i.e. FWSM which was considered to be a mixed-integer binary nonlinear programming type. This is a novel model of fuzzy multi-objective multi-period (seasonal) inventory model at which the objective of minimizing the required storage space is considered for the first time in addition to minimizing the total inventory costs.

In the second part, two situations were provided for formulating the proposed ISLA. First, a novel two-echelon multi-product multi-period inventory-supply chain problem was investigated for a facility location allocation problem at which the

capacity of the vendors' warehouses, the total available budget and the production capacity were limited. The products were sold under AUD discount policy and the distance among the buyers and vendors were considered to be Euclidean. Then, a distributor-retailer multi-item multi-period inventory problem was modeled for a facility location allocation problem where the shortages were allowed and in case of shortage, a fraction was lost sale for some products and a fraction was backordering for some other products. Products were purchased under AUD and IQD discount policies and the distance among distributors and retailers were defined to be Euclidean and Square Euclidean.

Numerical and computational examples were generated to assess the performance of the proposed algorithms. In order to solve the proposed models, meta-heuristic algorithms namely MOGA, PSO, SA, GA and MFOA were utilized. In addition, a design of experiment approach called Taguchi was applied to tune the parameters' algorithms as well. Furthermore, the algorithms were evaluated using case studies.

All algorithms have been coded on MATLAB R2013a and the codes have been executed on a computer with 3.80 GHz and 4 GB of RAM. Lastly, all the graphical and statistical analyses were performed in MINITAB 15.

1.5 Organization of the Thesis

There are seven chapters in this thesis, which are arranged as follows:

Chapter 1: In this Chapter, the background of study, problem statements, research objectives and research questions, and scope of the work were explained.

Chapter 2: This Chapter contains an overall literature review using the works which are relevant to inventory control problems, supply chain management, inventory-supply chain management and location allocation problem.

Chapter 3: This Chapter presents the detailed methodology applied to carry out the research objectives. This chapter also describes a brief explanation of various optimization algorithms used in this research study.

Chapter 4: This Chapter emphasizes on examining and comparing the results of MOPSO and MOGA on a multi-objective multi-product multi-period inventory control problem.

Chapter 5: A two-echelon multiple products, multiple periods inventory-supply chain model is derived for a facility location allocation problem. Two tuned meta-heuristic algorithms (PSO and GA) are used to solve the problems and then their results are compared to each other as well.

Chapter 6: In this Chapter, a distributor-retailer multi-item multi-period inventory supply chain problem is formulated in a location allocation problem where MFOA, PSO and SA are employed to optimize the problem.

Chapter 7: This Chapter proposes the conclusion of the research in addition to some recommendations for further study.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This Chapter is about of the review of the literature works related to multi-item multi-period inventory-supply chain and location allocation problems. Furthermore, the proposed works relevant to the meta-heuristic algorithms applied in inventory control, supply chain and location allocation in the literature are reviewed. Moreover, the literature review of the works applying the design of experimental method i.e. Taguchi approach is also performed.

2.2 Inventory Control Problem (ICP)

The inventory control problem is the problem encountered by a company which has to decide how much to order in each time-period to meet demand for its products. The problem can be formulated by using mathematical techniques of optimal control, dynamic programming and network optimization. The study of these techniques is part of inventory theory (Malakooti, 2013).

Planning for inventory order quantity and order time of the products are the important managerial decisions that affect the total costs associated with an inventory system. The most fundamental model that has been developed for this decision making is the economic order quantity (EOQ).

2.3 Inventory-Supply chain Problems (ISP)

2.3.1 Multi-objective ISP

For economic purposes, a wide range of problems is considered to be multi-objective with some constraints on what combinations of those objectives are attainable. For example, customer's demand for a variety of items is specified by the

process of maximization of the qualities conducted from those items, subject to a constraint based on how much salary is given by the employers to spend on those items and on the prices of those items.

A multi-objective optimization problem is an optimization problem that involves multiple independent objective functions which are in conflict with each other (Benson, 2000). In mathematical terms, a multi-objective optimization problem can be formulated as (Benson, 2000; Hwang & Masud, 1979):

$$\begin{aligned} & \text{Min } (f_1(x), f_2(x), \dots, f_k(x)) \\ & \text{s.t. } x \in X, \end{aligned} \tag{2.1}$$

where the integer $k \geq 2$ is the number of objectives and the set X is the feasible set of decision vectors. The feasible set is typically defined by some constraint functions. In addition, the vector-valued objective function is often defined as follows:

$$f : X \rightarrow \mathfrak{R}^k, f(x) = (f_1(x), \dots, f_k(x))^T$$

If some objective functions are to be maximized, it is equivalent to minimize their negative. An element $x^* \in X$ is called a feasible solution or a feasible decision. A vector $x^* = f(x^*) \in \mathfrak{R}^k$ for a feasible solution x^* is called an objective vector or an outcome.

2.3.1.1 Multi-product ISP

Instead of a single item, many firms, enterprises or vendors are motivated to store a number of products to enhance their business profitability. They are also motivated to attract customers to purchase several items simultaneously. Seasonal products are manufactured and stored in different periods such as sport clothing, the customers

usually prefer to have the matched clothes relevant to the sport they are doing. For example, one may wear a blue shirt with a blue short as a football player.

Ben-Daya and Raouf (1993) considered a multi-product, single-period inventory control problem with stochastic demand, for which the multi-periodic inventory control problem was investigated in depth for multiple products producing in multiple periods. Recently, (Wang & Xu, 2009) studied a multi-period, multi-product inventory control problem having several inventory classes that can be substituted for one another to satisfy the demand for a given reservation class. The weakness of the work was that shortages were not considered while formulating the model in uncertainty could be as its strength. A dynamic programming approach was employed to model the problem. Das et al. (2000) formulated a multi-item inventory model in which the demands were fixed and infinite replenishment (multi-period) under restrictions on storage area were considered, where costs included average shortage, and inventory investment cost. The weakness was that only backorder was considered and its strength was that a fuzzy version of multi-objective inventory was formulated.

Mohebbi and Posner (2002) studied an inventory system based on periodic review, multiple replenishment, and multilevel delivery. They assumed that the demand follows a Poisson distribution, that shortages were allowed. Considering lost sale only could be as its weakness while backordering was not proposed. Lee and Kang (2008) improved an inventory model to manage the inventory of a single product in multiple periods in a real case study in Taiwan where the weakness was that the model was provided only for a case study and was not applicable for other problems. Padmanabhan and Vrat (1990) developed a multi-item/objective inventory model having deteriorating items with stock dependent demand using a

goal programming method. Moreover, (Taleizadeh et al., 2013) proposed a multi-product inventory control problem with a stochastic replenishment period in which the demands were fuzzy numbers, and shortages were allowed to occur with a combination of backorders and lost sales. Considering the problem in both fuzzy and stochastic environments could be considered as its strength and ignoring some real constraints was its weakness. Chakraborty and Hasin (2013) studied a multi-objective multi-item multi-period inventory control problem (production planning) where the aim was to minimize the total costs with accordance to order quantities, labor levels, overtime, subcontracting and backordering levels, and labor, machine and warehouse capacity. Shortages were not considered. Ebrahimipour et. al. (2015) provided a multi-product multi-objective preventive maintenance scheduling problem in a multi-production line in which reliability of production lines, costs of maintenance, failure and downtime of system were calculated as multi-objective problem, and a variety of thresholds for available manpower, spare part inventory and periods under maintenance were used.

Yousefi et. al. (2012) developed a multi-item multi-objective joint replenishment inventory problem where the objectives were minimizing the total holding and ordering costs, and minimizing the transportation costs as the second objective. The problem was formulated wrongly while all the objectives were in the same unit and identical. Esfandiari and Seifbarghy (2013) considered a multi-objective inventory supply chain problem with multiple products under stochastic environment in which the supplier prices were linearly dependent on the order size of the items. A multi-objective multi-criteria supplier selection and inventory control problem with multiple products formulated as mixed integer programming was modeled by (Parhizkari et. al. 2013). Their proposed inventory objectives included minimizing the quality and offering price of suppliers, simultaneously. They did not used the

classical multi-objective problems as well as evolutionary algorithms to solve the problem. Moattar et. al. (2014) formulated an integrated multi-objective multi-product production and distribution planning in the presence of Manufacturing Partners where the goals were to minimize the total costs and maximize the quality level of the products supplied by distributors. The weakness was that shortages were not allowed.

Khalili-Damghani and Shahrokh (2014) studied a multi-objective multi-item multi-period production planning problem considered as a mixed-integer mathematical programming in which the aims were to minimize the total inventory costs, maximize customer services and maximize the quality of end products. Modeling the problem in fuzzy environment was the work strength since it would not be resolvable once the dimension of the problem increases. Gholamian et. al. (2015) considered a multi-objective multi-product multi-side multi-period mixed-integer nonlinear programming inventory supply chain problem under fuzzy environment. Gholamian et. al. (2015) presented a mixed-integer nonlinear model for a multi-objective multi-item inventory supply chain production planning under uncertainty where the objectives contained minimizing the total inventory costs, maximizing customer services and minimizing the rate of changes in human resource. The weakness of all the works reviewed in this section would be that all of them were formulated in a single-period.

2.3.1.2 Multi-period (seasonal) ISP

This work proposed a multi-period inventory model for multiple items. The multi-periodic inventory control problems have been investigated in depth in a wide range of works. Chiang (2003) investigated a periodic review inventory model in

which the period was partly long. The important aspect of his study was to introduce emergency orders to prevent shortages. He employed a dynamic programming approach to model the problem. Mohebbi and Posner (2002) investigated an inventory system with periodic review, multiple replenishment, and multilevel delivery. Assuming a Poisson process for the demand, shortages were allowed in this research, and the lost sale policy could be employed. The model was formulated only for the small-size problem with one item which could be considered as its weakness. Lee and Kang (2008) developed a model for managing inventory of a product in multiple periods. Their model was first derived for one item and then was extended for several products. Mousavi et. al. (2013) proposed a multiproduct multi-period inventory control problem under time value of money and inflation where total storage space and budget were limited. They solved the problem using two meta-heuristic algorithms, that is, genetic algorithm and simulated annealing. The problem was not considered for those companies building or extending their warehouses while the truck capacity and also shortages were not proposed. However, the problem was calculated the costs under inflation and currency fluctuation rates. Mirzapour Al-e-hashem and Rekik (2014) presented a multiproduct multi-period inventory routing problem, where multiple constrained vehicles distributed products from multiple suppliers to a single plant to meet the given demand of each product over a finite planning horizon. They only provided the problem with one plant which was rare in the real world company competition. Janakiraman et. al. (2013) analyzed the multiperiodic newsvendor problem and proposed some new results.

Pasandideh et. al. (2013) addressed a multi-product multi-period inventory control problem considered as mixed-integer nonlinear programming with storage and the number of packets capacity which was solved by Memetic and genetic

algorithms. Mousavi et. al. (2013) studied a multi-item multi-period mixed-integer, nonlinear inventory control problem in which shortages were allowed and the storage space and the order quantity were constrained. The aim was to find out the number of packets and order quantities so that the total costs became minimized. The objective of optimizing the total storage space for storing the items could be investigated. Li et. al. (2012) considered a multi-item multi-period capacitated dynamic lot-sizing problem where each item faced a series of dynamic demands, and in each period, multiple items shared limited production resources. In their research, shortages were allowed in the forms of backorder and lost sale. Mousavi et. al. (2013) formulated a mixed-integer nonlinear model for a seasonal multi-product multi-period inventory control problem in which the total budget and the storage space were limited. The aim was to minimize the total inventory costs including ordering, holding, purchasing and shortage costs. Pasandideh et. al. (2013) modeled a multi-item multi-period mixed-integer binary nonlinear inventory control problem where the items were packed into the pre-specified boxes. The shortages were not allowed as its weakness.

Inventory problems dealing with fashion products are usually modeled in a known number of periods. A few research works on multiperiodic inventory control problems have been conducted. Ahmed et. al. (2007) investigated a multi-period single-item inventory problem with linear cost, where the objective function was a coherent risk measure. The model could be extended for multi-item inventory problem. Sepehri (2011) formulated an integrated flow network and expanded it to a multi-period multi-product inventory control problem with the possibility of holding inventories in a multi-stage multimember cooperative supply chain. Their model did not consider shortages as an important part of the costs. Zhang et. al. (2009) presented some convex stochastic programming models for multi-period

inventory control problems where the market demand was random and order quantities needed to be decided before the demand was realized. As a strength of the work, the problem was formulated in uncertainty. Choi et. al. (2011) proposed a solution scheme for a periodic review multi-period inventory problem under a mean–variance framework.

In the literature of inventory seasonal items, there are different models with different definition of demand seasonality. In some works, a specific pattern (synchronized fluctuation) of seasonal demands were considered (Eppen & Iyer, 1997; Liu et al. 2017; Van Mieghem & Dada, 1999; Swinney et al., 2011; Jinzhao Shi et al., 2017). In many works in the literature published recently, seasonality has been considered to be multi-period where the demands values have been variable in different periods. These values have been generated randomly due to lack of benchmark in the literature (Dayarian et al., 2016; Diewert et al., 2009; Mogale et al., 2017; 2016; Saracoglu et al., 2014; Tanksale & Jha, 2017). Moreover, in the work proposed by (Costantino et al., 2016), it has been clearly mentioned that the seasonal demands can be random or stochastic.

2.3.1.3 Discounted ISP

Quantity discount strategies has attracted further attention because of its practical importance in purchasing and control of a product. It derives better marginal cost of purchase/production availing the chances of cost savings through bulk purchase/production. In supply chains, quantity discounts can be considered as an inventory coordination mechanism between a buyer and a supplier (Shin & Benton, 2004). Benton (1985) considered an inventory system having quantity discount for multiple price breaks and alternative lot-sizing policy. Maity (2011) developed a model for multi-item inventory control system based on breakable items, taking into

account all units (AUD) and incremental quantity discount (IQD) policies where the shortages were not allowed. In this thesis, a combination of AUD and IQD is used, which is rather similar to the work by (Sana & Chaudhuri, 2008). Sana and Chaudhuri (2008) extended an EOQ model based on discounts through the relaxation of the pre-assumptions associated with payments where the real world limitations as well as budget and truck capacities were ignored. Furthermore, (Taleizadeh et. al., 2011) considered a mixed integer, nonlinear programming for solving multi-product multi-constraint inventory control systems having stochastic replenishment intervals and incremental discounts for which a genetic algorithm was employed to find the near-optimum order quantities of the products. In their work, the uncertainty was considered as a real case; however, the algorithms' parameters were not tuned statistically. Recently, Mousavi et. al. (2013) improved the solution of a discounted multi-item multi-period inventory control problem for seasonal items, in which shortages were allowed, and the costs were calculated under inflation and time value of money. As a weakness, the supply chain and location-allocation parts of the work was not considered

Quantity discount policies are usually treated as an inventory coordination mechanism between buyers and suppliers (Shin & Benton, 2004). To name a few research works in this area includes (Chang & Chang, 2001) who used a linear programming relaxation based on piecewise linearization techniques to solve an inventory problem with variable lead-time, crashing cost, and price-quantity discount. The model was formulated and solve only for the problems with small-size. Further, (Taleizadeh et al., 2010) proposed a fuzzy mixed-integer nonlinear programming model for a multiproduct multi-chance constraints inventory problem with probabilistic period length and total discount under fuzzy purchasing price and holding cost and solved it by a hybrid meta-heuristic intelligent algorithm. Mousavi

et al. (2014a) investigated a multi-objective multi-item multi-period inventory control problem where IQD and AUD policies were considered. Besides, shortages were allowed and the discount rates were calculated as fuzzy values in the proposed model. The model compares two design of experiment methods i.e. Taguchi and Response Surface Methodology (RSM) to tune the algorithms' parameters where the Taguchi performance was found to be better than RSM to optimize values of the parameters. Chen et al. (2015) proposed an integrated multi-objective inventory supply selection with some supplier's discount policy.

2.3.1.4 Fuzzy ISP

In general, many of the variables, resources, and constraints used in a decision-making problem are considered either deterministic or stochastic. However, real-life scenarios demand them to be imprecise, which are, uncertainty is to be imposed in a non-stochastic sense. According to (Maiti & Maiti, 2008), a business may start with some warehouse spaces in an inventory control problem. However, due to unexpected demands, some additional storage spaces may need to be added. These added spaces are normally imprecise in nature that should be determined. Moreover, one of the weaknesses of many current inventory models is that they have not optimized the storage space required for the items ordering for satisfying the demands in uncertainty. Therefore, this study formulates a novel inventory problem to optimize both objectives of the required storage space and the total inventory costs, simultaneously. While the weights of these objectives are unspecified and uncertain for the decision makers, they are considered fuzzy numbers in this thesis. Maity and Maiti (2008) formulated an optimal production strategy for an inventory control system of deteriorating multi-items under a single owner based on resource constraints under inflation and discounting in a fuzzy environment. Recently, (Chen

& Ho, 2013) investigated a newsboy inventory problem in a fuzzy environment by analyzing the optimal inventory policy for the single-order newsboy problem considering fuzzy demand and quantity discounts. Moreover, (Guchhait et al., 2013) modeled an inventory control problem under discount based on fuzzy production rate and demand. Mousavi et al. (2014a) addressed an inventory control problem with fuzzy discount rates.

In this study, a triangular fuzzy is used to show the fuzzy membership functions of the weights of the objectives and a α -cut is also applied to convert these fuzzy variables to crisp ones. These two approaches have been considerably applied in Inventory and supply chain fields in the literature, recently. Priyan et al. (2014) used Triangular and trapezoidal fuzzy numbers for modeling a fuzzy multi-period EOQ inventory problem where α -cut approach was applied to convert the fuzzy numbers to crisp. Kao and Hsu (2002) formulated a fuzzy lot-sizing reorder point inventory problem in which demands were considered fuzzy numbers. In their work, α -cut of the fuzzy demand was used to construct the fuzzy total inventory cost for each inventory policy. Sadi-Nezhad et al. (2011) investigated a multi-item multi-period inventory control problem with fuzzy setup cost, holding cost and shortage cost where triangular and α -cut approaches were used to show and convert fuzzy numbers, respectively. Handfield et al. (2009) developed a fuzzy (Q, r) inventory model in a supply chain problem where demand, lead time, supplier yield and penalty costs were considered to be fuzzy numbers. Both triangular and α -cut methods were utilized to depict and construct fuzzy numbers. De and Sana (2013) a fuzzy α -cut technique was incorporated for fuzzy optimization in an EOQ inventory control problem where the order quantity, shortage quantity and the promotional index were triangular fuzzy decision variables.

In order to combine the fuzzy objectives into one objective, FWSM is used where weighted sum method is the most common method for multiobjective optimization problems (Kim & Weck, 2006). There are a lot of works in the literature used FWSM (Cheng et al., 2013; Ma et al., 2017; Majidi et al.2017; Stojiljković, 2017; Su, 2017), and also some works published in Supply chain and Inventory recently (Lee, 2017; Maity & Maiti, 2008; Mousavi et al., 2014b).

Table 2.1 shows the literature review of the works reviewed in this work in which DOE is an abbreviation of term "design of experiments."

Table 2.1: Literature review of some of the works related to Section 2.3.1

References	Multi-product	Multi-period	Fuzzy Multi-objective	Discount policy	Solving Methodology	Shortages	DOE
(Chiang, 2003)	-	√	-	-	B&B	-	-
(Mohebbi & Posner, 2002)	-	-	-	-	Level-crossing	√	-
(A. H. Lee & Kang, 2008)	-	√	-	IQD	Numerical methods	-	-
(Mousavi et al., 2013)	√	√	-	IQD	GA	-	-
(Mirzapour Al-e-hashem & Rekik, 2014)	√	√	-	-	CPLEX	-	-
(Janakiraman et al., 2013)	-	√	-	-	-	√	-
(Shin & Benton, 2004)	-	-	-	IQD&AUD	Simulation	-	-
(Benton, 1985)	-	-	-	AUD	Simulation	-	-
(P. Abad, 1988)	-	-	-	IQD	Numerical methods	-	-
(P. L. Abad, 1988)	-	-	-	IQD	Numerical methods	-	-
(Maiti & Maiti, 2008)	√	-	-	IQD&AUD	GA	-	-
(Sana & Chaudhuri, 2008)	√	-	-	AUD	Numerical methods	-	-
(Taleizadeh et al., 2010)	√	-	-	IQD&AUD	GA	√	-
(S.-P. Chen & Ho, 2013)	-	-	-	IQD	Yager ranking	-	-
(Khan et al., 2014)	-	-	-	IQD	Excel macro	-	-
(Taleizadeh et al., 2013)	-	-	√	-	TOPSIS & GA	√	-
(S.-H. Huang & Lin, 2010)	√	-	-	-	ACO	-	-
(K. Li et al., 2014)	-	-	-	-	Tabu search & Lagrangian	-	-
(Taleizadeh et al., 2008)	-	√	-	-	Tabu search	-	-
(Taleizadeh et al., 2009)	-	-	-	IQD	Goal programming & GA	√	-
(Pasandideh et al., 2014)	√	-	-	IQD	GA & Fuzzy simulation	√	-

Table 2.1, Continued

(Taleizadeh et al., 2009)	√	-	-	-	GA	√	-
(Taleizadeh et al., 2011)	√	-	-	-	SA & GA	√	-
(Yaghin et al., 2013)	-	-	√	-	TOPSIS & GA	√	-
(Aarts & Korst, 1988)	√	-	-	IQD	fuzzy simulation	√	-
(Dorigo & Birattari, 2010)	√	-	-	-	Harmony search	√	-
(Kennedy & Eberhart, 1995)	√	-	-	-	Bee colony & PSO	√	-
(Y.-R. Chen & Dye, 2013)	√	-	√	-	Fuzzy programming algorithm	-	-
(Coello Coello & Lechuga, 2002)	-	√	√	-	Fuzzy method	-	-
This research	√	√	√	√	PSO & GA	√	Taguchi

2.3.2 Single-objective ISP

Advances in SCM software and data warehousing practices, which enable data sharing through Electronic Data Interchanges (EDI), have helped in the development of coordinated supply chains (Tan, 2001). Supply chain optimization involves both strategic decisions of facility location, and tactical decisions of inventory. Traditional supply chain optimization models in the literature treat location and inventory decisions separately (Wang & Yin, 2013). An integrated Supply Chain Management (SCM) strategy allows companies to increase efficiency and decrease waste. More specifically, successful implementation of an integrated SCM model results in savings in energy and fuel in addition to the elimination of redundant activities; all of these benefits translate into money savings within the companies (Mentzer et al., 2001; Spekman et al., 1998). However, ignoring interaction between long-term decisions of location and short-term decisions of inventory can lead to sub-optimality. Furthermore, companies are under intense pressure to cut product and material costs while maintaining a high level of quality and after-sale services. Achieving this starts with supplier selection. Therefore, an efficient supplier selection process needs to be in place and of paramount importance for successful SCM (Chan et al., 2008). Inventory management is one of the important scopes in SCM and many academic communities have presented various strategies.

In this work, the design of a two-echelon vendor-buyer supply chain network for a multi-product multi-period inventory problem is investigated. The two-echelon means the supply chain network includes only two members i.e. vendors and buyers. It will include three members if the supply chain network is a three-echelon one. In recent years, numerous studies have been carried out on supply chain and inventory problems considering the two-echelon design. In Sadeghi et al. (2013b), a

constrained the two-echelon multi-vendor multi-retailer single-warehouse supply chain was developed, in which both the space and the annual number of orders of the central warehouse were limited. A coordination model of the joint determination of order quantity and reorder point variables was proposed by (Chaharsooghi & Heydari, 2010). The research decentralized supply chain consisting of one buyer and one supplier in a multi-period setting. The design of a two-echelon production distribution network with multiple manufacturing plants, distribution centers and a set of candidate warehouses was considered by (Cardona-Valdés et al., 2014). The study took into account a multi-objective version of supply chain in stochastic environment and a Tabu search algorithm was used to solve the problem. In Bandyopadhyay and Bhattacharya (2013) a NSGA II was proposed to solve a tri-objective problem for a two echelon serial supply chain. The objectives were: (1) minimization of the total cost of a two-echelon serial supply chain and (2) minimization of the variance of order quantity and (3) minimization of the total inventory. Ghiami et al. (2013) studied a two-echelon supply chain model for deteriorating inventory was investigated, in which the retailer's warehouse had a limited capacity. The proposed system included one wholesaler and one retailer, which aimed to minimize the total cost. Sadeghi et al. (2014a) developed a bi-objective vendor managed inventory model in a supply chain with one vendor and several retailers, in which the determination of the optimal numbers of different machines, working in series to produce a single item was considered.

2.3.2.1 Inventory-Supply Chain Management Location Allocation (ISLA)

Most real-world problems in industries and commerce are studied using a single-objective optimization model. The assumption that organizations always seek to minimize cost or maximize profit rather than make trade-offs among multiple objectives has been used extensively in the literature. In this regards, classical

inventory models have been developed under the basic assumption that a single product is purchased or produced. There are many works have been performed on ICP in different areas of the study in the recent decade. Das and Maiti (2007) studied a single period newsboy type inventory problem for two substitutable deteriorating items where the resource were constrained. The demands of the customers were probabilistic and lot sizes were also random. The aim was to maximize a single objective profit function with the space constraint. Dey et al. (2008) developed a finite time horizon inventory problem for a deteriorating item with two separate warehouses including an own warehouse of finite dimension and a rented warehouse where lead-time was considered to be interval-valued lead-time and inflation and time value of money were calculated into the model.

Dutta and Chakraborty (2008) proposed a novel way to develop several strategies for single-period inventory models in a real decision-making situation. They provided their model under uncertainty in the demands of customers where the demands were described by imprecise terms and formulated by fuzzy sets. The aim was to find out the optimal order quantity that will maximize the total expected profit after the end of the season. Chen et al. (2008) investigated a classical inventory single-period newsboy model with fixed life cycle. The model was extended with reusable and imperfect products where the objective was to find the optimal order quantities in cases with or without corporation. Liu and Luo (2009) modeled a risk-averse newsvendor problem with return policy under the conditional value-at-risk criterion where the combination of the expected profit and conditional value-at-risk as the objective function was chosen. Chen et al. (2011) addressed a dynamic pricing problem of a single-item, make-to-stock production system at which the customer demands arrival was based on Poisson processes with changeable rates depending on selling prices. Additionally, product-processing times follow an Erlang

distribution where the aim is to identify a dynamic control policy which decided production and adjusted the price to maximize the long-run total profit under discount policy.

The optimal control of a capacitated periodic-review make-to-stock inventory control problem was addressed by (Zhou et al., 2011) at which production capacity was limited and demands arrived in different classes. The goal was to optimize the total inventory costs under discount policies. Peng et al. (2010) studied a finite horizon, single-product, multi-period model at which the decisions related to pricing and inventory had to be made at the beginning of each period at the same time. The aim was to maximize the expected profit in the whole planning horizon. A single-item single-period inventory control problem with discrete customer demands was described by (Behret & Kahraman, 2012) at which the demand rates were considered to be a stochastic variable while inventory costs including holding cost and shortage cost were imprecise and represented by fuzzy numbers. Choudhary and Shankar (2011) modeled a single product multi-period procurement lot-sizing problem formulated in an integer linear programming approach which is procured from a single supplier considering rejections and late deliveries AUD policy.

Two types of inventory lot-sizing problems including the production planning and control of a single product involving combined manufacturing and remanufacturing operations have been investigated by (Zouadi et al., 2013). Xiao et al. (2012) discussed an item assembly inventory planning and control system in a manufacturing environment at which the process of returning end-of-life item was considered to be stochastic in terms of arriving demand, quality and quantity. Paul et al. (2013) proposed a risk management approach for solving an inventory control problem for imperfect products with demand uncertainty and process reliability in which a non-linear constrained optimization model was modeled. The main

objective was to maximize the graded mean integration value of the total expected profit. Yang et al. (2014) addressed a single-product multiple review inventory control problem with finite-horizon and setting certain/uncertain supply chain capacity where in different periods the customer demands had random variables. Wang et al. (2015) studied a single-period single-item/multi-item inventory newsboy control problem in random and uncertain environment. Hnaien et al. (2015) proposed a single-period inventory control problem of an assembly system for one type of the finished item in which lead times and the customer demands had stochastic values.

2.3.2.2 Location-Allocation Problem (LAP)

In a location allocation (LA) problem, a number of new facilities are placed in between a number of specific customers in a feasible area such that the total transportation cost from facilities to customers is minimized. The location allocation model was proposed by (Cooper, 1963) and extended by numerous researchers such as (Harris et al., 2014; Hosseinienezhad et al., 2013; Mousavi & Niaki, 2012; Mousavi et al., 2013; Willoughby & Uyeno, 2001). Furthermore, (Logendran & Terrell, 1988; Sherali & Rizzo, 1991; Carrizosa et al., 1995; Carrizosa et al., 1998; Zhou, 2000), were some of the researchers who extended the LA problem in stochastic environments. Zhou and Liu (2003) studied a capacitated LA problem with stochastic demands and deterministic locations. They employed three types of stochastic programming to model the problem. Zhou and Liu (2007) considered a capacitated LA problem with fuzzy demands in which the customers' locations were deterministic. Wen and Iwamura (2008a) proposed a fuzzy facility LA model under the Hurwicz criterion. Wen and Iwamura (2008b) utilized a facility LA model in random fuzzy environment. They used the (a,b) -cost minimization under the Hurwicz criterion to formulate the problem. Abiri and Yousefli (2010) proposed an

application of the probabilistic programming approach to model the fuzzy LA problem where demands were fuzzy and locations were deterministic.

There are many works in the literature performed on inventory and supply chain with location allocation problems. Ghodrattnama et al. (2015) developed a multi-objective supply chain location allocation problem in which the objectives were to minimize total transportation and installation costs, minimize service time, tardiness and flows containing raw materials and also minimize total greenhouse gas emitted by transportation mode. Wang and Lee (2015) improved a capacitated facility location-allocation problem for a multi-echelon supply chain problem to formulate a stochastic model in which the aim was to maximize the total profit. Arabzad et al. (2015) presented a multi-objective facility location-allocation supply chain problem with multiple suppliers, multiple products, multiple plants and multiple customers. The objectives were minimizing the total supply chain costs and minimizing total deterioration rate caused by transportation alternatives. Karmaker and Saha (2015) investigated a multi-staged location-allocation fuzzy decision making supply chain problem.

2.3.2.3 Inventory-Supply Chain Location-Allocation without shortages (ISLAWOS)

This work considers a supply chain network in which several vendors (manufacturers) are considered to be located in a certain area between numerous buyers who own the warehouses with the limited capacity. Furthermore, the objective is to find the optimal quantity (allocation) that each buyer orders from the vendors. In a location allocation problem, several new facilities are located in between a number of pre-specific customers in a determined area such that the total transportation cost from facilities to customers is minimized. In supply chain

management, a number of studies have been probing the location allocation problem. Abolhasani et al., (2013) optimized a class of supply chain problems, known as multi-commodities consumer supply chain problem, where the problem considered to be a production-distribution planning category. It aimed to determine the facilities location, consumers' allocation and facilities configuration to minimize the total cost of the entire network.

Shahabi et al. (2013) developed mathematical models to coordinate facility location and inventory control for a four-echelon supply chain network consisting of multiple suppliers, warehouses, hubs and retailers. Wang and Yin (2013) investigated an integrated supply chain optimization problem in which the optimized facility locations, customer allocations, and inventory management decisions were considered when the facilities were subject to disruption risks. Diabat et al. (2013) considered a closed-loop location-inventory problem with forward supply chain consisting of a single echelon where the distribution centers had to distribute a single product to different retailers with random demands. Furthermore, (Ahmadi-Javid & Seddighi, 2013) developed a location-routing problem in a supply-chain network considering a producer–distributors set which produced a single commodity and distributed it to a set of customers.

In the inventory control problems, the vendors sell their products under some discount policies in order to attract and encourage the customers to increase purchase. AUD is one of most common policies that has been taken into account in the literature recently. Mousavi et al. (2013) modeled a seasonal multi-product, multi-period inventory control problem in which the inventory costs were obtained under inflation and all-unit discount policy. A multi-item multi-period inventory control problem with all-unit and/or incremental quantity discount policies under limited storage capacity was developed by (Mousavi et al., 2013). Recently, a

deteriorating multi-item inventory model with price discount and variable demands via fuzzy logic under all unit discount policy has been investigated by (Chakraborty et al., 2013). Chen and Ho (2013) analyzed the optimal inventory policy for the single-order newsboy problem with fuzzy demand and quantity discounts. Furthermore, an integer linear programming approach was used in (Choudhary & Shankar, 2011) to solve a multi-period procurement, lot-sizing problem for a single product that was procured from a single supplier, considering rejections and late deliveries under all-unit quantity discount environment.

In the recent decades, meta-heuristics algorithms have attracted the attention of many researchers in order to optimize different and complex problems in various engineering and science domains. The PSO algorithm is a popular and perhaps the most widely used meta-heuristic algorithms that was first introduced by (Eberhart & Kennedy, 1995). PSO is a population-based stochastic meta-heuristic algorithm that is inspired by the social behavior of bird flocking or fish schooling. PSO is a meta-heuristic that requires few or no assumptions on the problem being optimized and can search very large and complex spaces of candidate solutions. PSO can therefore be utilized on optimization problems that are partially irregular and noisy over time (Gigras & Gupta, 2012). This algorithm has been used for solving the inventory and supply chain problems in recent years. Mousavi et al. (2013) applied a PSO algorithm to solve a multi-product multi-period inventory control problem where the shortages (Backorder and Lost sale) were allowed.

In the multi-objective version, (Latha Shankar et al., 2012) modelled a single-product for four-echelon supply chain architecture consisting of suppliers, production plants, distribution centres and customer zones. They used a PSO algorithm to solve the problem. Bozorgi-Amiri et al. (2012) investigated a relief chain design problem for which, not only demands but also supplies and the cost of

procurement in addition to transportation were considered as the uncertain parameters. Furthermore, the model considered uncertainty for the locations solving using the PSO algorithm. Park and Kyung (2013) proposed a method to optimize both the total cost and order fill rates in a supply chain using the PSO method. They automatically adjusted the initial order quantities of all tiers involved in a supply chain by considering information quality level, which was determined by the degree of availability of lead time history data.

In order to optimize the proposed multi-product multi-period supply chain problem and determine the location of the vendors in a specific area among the buyers, a modified particle swarm optimization (MPSO) is utilized in this work.

2.3.2.4 Inventory-Supply Chain Location Allocation with shortages (ISLAWS)

Business requirements will change over time. This can be due to mergers and acquisitions, entrance to new markets, expansion of product ranges, or indeed changes to the regulatory environment. In the contemporary business environment, competition is no longer between organizations. The necessity to coordinate several business partners, internal corporate departments, business processes, and diverse customers across the supply chain gave rise to the field of SCM (Turban et al., 2009). The current business environment has become unpredictable with the consequence that the emphasis placed on the role played by logistics and SCM has been continuously increasing: the environment is becoming increasingly complex and competitive (Gebennini et al., 2009). The value of inventory is approximately 14% of gross domestic product in the United States, while annual transportation and warehousing expenses average approximately nine percent of gross domestic product (Wilson, 2005). Retail companies in the US spend approximately \$14 billion per year on inventory interest, insurance, taxes, depreciation, obsolescence,

and warehousing. Their logistics activities account for 15–20% of the total cost of finished goods (Menlo, 2007). With such a huge logistics investment, it is important to make sound decisions for facility locations and inventory allocation in a supply chain. The design and management of supply chain network in today's competitive business environment is one of the most important and difficult problems that managers face (Tsao & Lu, 2012).

(a) Multi-product ISLAWS

In this sub-section, the works relevant to multi-product ISLAWS in the literature were reviewed. The bulk of the literature has focused on two-level supply chains that supply a single product. Nowadays, for enhancing the power of the competition in the market, more companies prefer to produce and handle various items under integrated inventory systems. Karakaya and Bakal (2013) analyzed a decentralized supply chain with a single retailer and a single manufacturer where the retailer sold multiple products in a single period. The products differed in terms of a limited number of features only. Diabat et al. (2009) studied an integrated inventory and multi-facility location problems with single supplier and multiple retailers under risk pooling. They employed a genetic algorithm to find a near-optimum solution of the problem. Diabat et al. (2013) specified the location of warehouses and inventory policies at the warehouses and retailers in a nonlinear mixed-integer model of multi-echelon joint inventory-location problem, where a Lagrangian relaxation-based approach was applied to solve their problem. Ganesh et al. (2013) analyzed the value of information sharing using a comprehensive supply chain that has multiple levels, may have different degrees of information sharing, and supplies multiple products that may have different levels of substitutability and whose demands could be correlated to different degrees. Tsao and Sheen (2012) considered a multi-item supply chain with a credit period and

weight freight cost discounts in which the retailer bore the freight costs, but the freight carrier provided freight-transport discounts that were positively related to the weight of the cargo transported.

(b) Multi-period ISLAWS

In today's business world, time is a very important factor for managing the bulk of the products in supply chains especially when the process is involved in a number of time-periods. In this study, a multiple period multiple product supply chain integrated with a location-allocation problem is formulated. Considering a supply chain model with multiple periods particularly for those products manufacturing in different periods is becoming an interesting issue among the researchers. Ramezani et al. (2014) addressed the application of fuzzy sets to design a multi-product, multi-period, closed-loop supply chain network. The presented supply chain included three objective functions: maximization of profit, minimization of delivery time, and maximization of quality. Zhang (2013) proposed a model and solution method for coordinating integrated production and inventory cycles in a manufacturing supply chain involving reverse logistics for multiple items with finite multiple periods. Mousavi et al. (2013) modeled a capacitated multi-facility location allocation problems with stochastic demands and customer locations in which the distances between customers and facilities were considered to be Euclidean and Square Euclidean. In addition, a capacitated multi-facility location allocation problem with fuzzy demands and stochastic customers' locations were studied by (Mousavi & Niaki, 2013) where Euclidean and Square Euclidean functions were utilized to formulate the problem. Mousavi et al. (2013) developed a mixed binary integer mathematical programming model for ordering seasonal items in multi-item multi-period inventory control systems in which orders and sales occurred in a given season under interest and inflation factors. Rodriguez et al. (2014) proposed an

optimization model to redesign the supply chain of spare part delivery under demand uncertainty from strategic and tactical perspectives in a planning horizon consisting of multiple periods. Moreover, long-term decisions involve new installations, expansions, and elimination of warehouses and factories handling multiple products. These decisions also correspond to which warehouses should be used as repair workshops in order to store, repair, and deliver used units to customers. Omar (2013) developed and analyzed a game theoretic model for revenue-dependent revenue sharing contracts wherein the actual proportion in which the supply chain revenue was shared among the players would depend on the quantum of revenue generated. The aim was to understand why revenue-dependent revenue sharing contracts were (or not) preferred over revenue-independent contracts. He also examined if supply chains could be coordinated over multiple periods using both types of revenue sharing contracts.

(c) Discounted ISLAWS

In the addressed supply chain model, the distributors order the multiple products in multiple periods under a couple of discount policies which are all-unit and incremental quantity discount where the products are delivered in known packets with a certain number of items. Lin (2013) incorporated overlapped delivery and imperfect items into the production–distribution model. The model improved the observable fact that the system might have experienced shortage during the screening duration and considered quantity discount. Tsao and Lu (2012) addressed an integrated facility location and inventory allocation problem considering transportation cost discounts. Their study considered two types of transportation discounts simultaneously: quantity discounts for inbound transportation cost and distance discounts for outbound transportation cost. Li and Liu (2006) developed a model for illustrating how to use quantity discount policy to achieve supply chain

coordination. A supplier-buyer system selling one type of product in multiple periods and probabilistic customer demand was considered. They also showed that if both the buyer and the supplier can find a coordination mechanism to make joint decisions, the joint profit is more than the sum of their profits in the decentralized decision situation. Mousavi et al. (2014) employed a PSO algorithm to solve a inventory-location allocation problem in which AUD strategy was applied for purchasing costs and Euclidean and square Euclidean distances were used to compute distances between vendors and buyers. Huang et al. (2014) considered a supply chain in which one manufacturer sells a seasonal product to the end market through a retailer. Faced with uncertain market demand and limited capacity, the manufacturer can maximize its profits by adopting one of two strategies, namely, wholesale price rebate or capacity expansion. In the former, the manufacturer provided the retailer with a discount for accepting early delivery in an earlier period. Chang and Chou (2013) considered a single product coordination system using a periodic review policy, where participants of the system, including a supplier and one or more heterogeneous buyers, are involved over a discrete time planning horizon in a manufacturing supply chain with seasonal demands.

2.3.3 Lot-Sizing ISLA (LSISLA)

The Lot-sizing can be defined as the quantity of a product demanded for delivery on a specific period (or multi-period) or produced in a single production run. In fact, the total items ordered for manufacturing is defined as lot-sizing. Based on (Brahimi et al., 2017), planning for ordering the quantity of a single item in different periods with different demand values for manufacturing would be considered as lot-sizing where the aim is to obtain the production amount of items to satisfy all the demands. In this research, the proposed models can be linked to lot-sizing problems.

In the literature of lot-sizing inventory problems, there has been a wide range of interests with producing due to their large potentials in reverse logistics. The three recent works reviewed the modelling of single item and multiple items lot-sizing and reverse logistics inventory systems have been studied by (Aloulou et al., 2014; Bazan et al., 2016; Brahimi et al., 2017), accordingly. A variable neighborhood heuristic algorithm was applied by (Sifaleras & Konstantaras, 2017) to solve a multi-item multi-period dynamic lot-sizing problem in the reverse logistics where the items were recoverable. Due to no benchmark in the literature, they generated data randomly to evaluate the algorithm on the problem.

Carvalho and Nascimento (2016) modeled a multi-plant multi-item multi-period lot-sizing problem where plants produced the same items with single machine for their own demands in different periods. They used a novel Lagrangian heuristic algorithm in addition to several other heuristics to solve the problem where a number of numerical examples were generated due to non-availability of the benchmark in the literature. Vital Soto et al. (2017) formulated a mixed-integer nonlinear programming for a multi-period inventory lot-sizing problem with supplier selection where the shortages were allowed and AUD and IQD policies were given by suppliers. To solve the problem, an evolutionary algorithm as well as a linear programming driven local search were utilized validated on some case studies and generated examples. A multiple item capacitated lot-sizing and scheduling problem was modeled by (Masmoudi et al., 2016) in a flow-shop system under energy consideration to formulate a mixed-integer linear programming. A fix-and-relax heuristic was applied to solve the problem while the capacitated lot-sizing problem was completely NP-hard. They generated some numerical examples in different sizes where the results were compared with different methods graphically, statistically and using CPU time.

Jiao et al. (2017) studied a multi-item multi-period (finite horizon) incapacitated inventory lot-sizing problem under uncertainty and AUD policy in which the shortages were allowed. The aim was to minimize the total inventory costs including ordering and holding costs where a polynomial time algorithm was applied to obtain the optimal order quantities due to the complexity of problem. The 12 generated different problems with different sizes were proposed with T ranging 3 to 9 different periods. Parsopoulos et al. (2015) used several metaheuristic algorithms as well as Differential Evolution, PSO and Sm variants to determine the optimal solutions in a multi-item multi-period lot-sizing inventory problem for manufacturing the items where some return items should have remanufactured. A mixed-integer programming model was formulated by (Boonmee & Sethanan, 2016) for a multiple levels multi-product multi-period capacitated lot-sizing and scheduling problem where shortages were allowed all in lost sale. A modified version of PSO called GLNPSO was used to solve the proposed model in which a traditional PSO was applied to validate the results of GLNPSO. They evaluated the algorithms' performances on a case study in Thailand and also on 12 generated numerical problems with different sizes where the results were compared graphically, statistically and using CPU time.

Noblesse et al. (2014) investigated a multi-item multi-period lot-sizing for both inventory and production systems where a markov chain analysis was applied to find out the lead time distribution. Absi et al. (2013) formulated a multi-item capacitated lot-sizing inventory problem where some of the demands were lost to be considered as shortage costs. In order to solve the problem, a Lagrangian relaxation of the capacity limitations was adopted where an adaptation of the O(T²) dynamic programming algorithm was also derived for solving the sub-problems. Absi et al. (2009) presented a model of multiple products multi-period capacitated lot-sizing problem in which setup times, safety stocks and shortages all in lost sale

were considered. Due to the complexity of the problem, a Lagrangian relaxation of the capacity constraints in addition to a dynamic programming algorithm and an upper bound algorithm were used to solve the problem. They generated 72 different problems with different sizes to evaluate the solution methods performance. Cárdenas-Barrón et al. (2015) formulated a multi-item multi-period lot-sizing inventory with supplier selection problem at which a hybrid algorithm integrating ROA and a heuristic was applied to solve the model. To validate the performance of the algorithm, a CPLEX solver was used where 150 numerical examples generated by (Basnet & Leung, 2005) including 75 small-sized, 30 medium-sized and 45 large-sized were proposed to evaluate the algorithm performance.

2.4 Solution Methods for Solving ISLA and LSISLA

In most inventory systems, due to some unexpected matters, supply chain networks face shortages to satisfy customers' demands. This work investigates a supply chain model in which the retailers, as the customers of the distributors, with variable demands in different periods may face shortages for which a proportion is lost sales and the rest is backorders. Mirzapour Al-e-hashem et al. (2013) developed a stochastic programming approach to solve a multi-period multi-product multi-site aggregate production planning problem in a green supply chain for a medium-term planning horizon under the assumption of demand uncertainty where shortages were penalized by a general multiple breakpoint function, to persuade producers to reduce backorders as much as possible. Singh and Saxena (2013) presented a closed loop structure with remanufacturing for decaying items under shortage rates. The model was considered for single item with two different quality standards, where amount of product was collected from the user. After the collection process, a ratio of the collected items was to be remanufactured. Furthermore, a salvaged option was considered within a certain structure.

In order to generate the feasible solutions those satisfy all the constraints of the proposed models in this study, penalty function technique is used in which a big value will be added to the objectives values as a penalty for the infeasible solutions. There are many works in the literature used penalty function technique to make feasible solutions. In fact, using penalty function approach is a really common method in the literature to generate feasible solutions satisfying all the constraints (Chambari et al., 2013; Fattahi et al., 2015; Hajipour & Pasandideh, 2012; Hajipour et al., 2014; Mohammadi, 2015; Mohammadi et al., 2015; Pasandideh et al., 2013; Rahmati et al., 2014; Rahmati et al., 2013; Sadeghi et al., 2014; Sadeghi & Niaki, 2015; Sadeghi, et al., 2014a; Sadeghi et al., 2014b).

2.4.1 PSO and GA

PSO and GA are the most common algorithms in the literature used to solve inventory, supply chain and location allocation problems where many researchers improved their problems by applying these two algorithms. Roozbeh Nia et al. (2013) developed a multi-item economic order quantity model with shortage under vendor managed inventory policy in a single vendor single buyer supply chain. The model explicitly included warehouse capacity and delivery constraints, bounded order quantity, and limited the number of pallets. A multi-product multi-chance constraint joint single-vendor multi-buyers inventory problem was considered by (Taleizadeh et al., 2012) in which the demand followed a uniform distribution, the lead-time was assumed to vary linearly with respect to the lot size, and the shortage in combination of backorder and lost-sale was assumed. They solved their problem using PSO and GA algorithms where the three different numerical examples were generated to evaluate the performance of the algorithms on the model. Mousavi et al. (2016) used multi-objective version of GA and PSO called the NSGA-II, MOPSO and NRGa to solve a multi-objective multi-item multi-period inventory

problem under inflation and time value of money. Three different numerical illustrations were generated to validate the performance of the proposed algorithms where Taguchi approach was used to tune the algorithms' parameters as well. Some graphical, statistical and CPU time analysis were created to compare the results of the algorithms.

Ahmadzadeh et al. (2017) utilized GA, ICA and FA algorithms in inventory-location and supply chain problem where a Taguchi method was designed to set the parameters of the used algorithms. The authors generated 30 different problems with different sizes using a uniform distribution such as the method applied in this research. Sadeghi and Niaki (2015) solved a multi-objective fuzzy inventory-supply chain problem using multi-objective version of GA i.e. NSGA-II and NPGA algorithms in which Taguchi approach was designed to adjust the algorithms' parameters. Dye and Ouyang (2011) used a PSO to obtain the optimal order quantity in a deteriorating multi-item multi-period lot-sizing inventory and pricing problem where the demands fluctuated in each period. They tested the algorithms on the three numerical examples generated randomly with different sizes where all the results were compared to each other statistically and graphically for the algorithms. Kundu et al. (2017) studied a multi-item multi-period economy production quantity problem under discount policy in each period in which the demand values were considered fuzzy numbers, where a fuzzy differential equation and the α -cut approaches were used. A combination of PSO and GA was employed to obtain the optimal solutions to be evaluated on several generated numerical problems. In the work presented by (Samal & Pratihari, 2014), a fuzzy multi-period EOQ inventory model with and without shortage consideration and variable demands was derived, in which PSO and GA were applied to solve the problem. Triangular and Gaussian

fuzzy membership function were used to show the fuzzy variables where several case studies and the generated data were utilized to evaluate the algorithms.

Soleimani and Kannan (2015) proposed a multi-item multi-period multi-echelon inventory-supply chain and location-allocation problem in which PSO and GA algorithms were conducted to solve the model. Some numerical examples including 10 small-size and 11 large-size problems were generated randomly where the algorithms were run on the problems and their results were compared to each other graphically, statistically and using CPU time. Hajipour et al. (2014) used MOGA, MOHS and SA to solve a multi-server location-allocation problem in which 25 different problems with different sizes were generated. The performance and quality of the algorithms were compared to each other using some statistical and graphical approaches. Fattahi et al. (2015) applied five metaheuristic algorithms as well as NSGA-II, NREGA, MOVDO, MOICA and MOPSO to solve a multi-objective multi-product (r,Q) inventory control problem where Taguchi approach was applied to tune the algorithms parameters. They generated 10 different problems with different sizes to evaluate the algorithms performance while the quality of the algorithms was compared to each other statistically and graphically.

There are also numerous works in the literature optimizing lot-sizing problems using GA and PSO. In (Mohammadi, 2015; Mohammadi et al., 2015), a multi-item multi-period capacitated lot-sizing inventory and scheduling problem consisted of several suppliers, multiple plants and distribution centers was modeled. In the work, metaheuristic algorithms as well as PSO, GA, SA and ABC were used to solve 30 generated numerical examples in small, medium and large sizes each 10 instances where Taguchi method was employed to tune the proposed algorithms. The results were compared graphically and statistically and also using CPU time. Karimi-Nasab et al. (2015) formulated a mathematical modeling of a multi-item multi-period lot

sizing and scheduling problem in a job shop where dealt with a main realistic assumption of flexible machines to change the working speeds, known as process compressibility. The shortages were allowed where demand values were different in the different periods. A PSO algorithm was applied to solve the problem in which a Lingo solver was used to compare the performance of PSO on several case studies in real industry. Duda (2017) utilized a GA algorithm and VNS to solve a multi-item capacitated lot-sizing multi-facility problem under setup times. They generated 10 instances in small-size and 10 instances in large-size to evaluate the algorithms' performance where statistical and graphical approaches were used to compare the performances of the algorithms. Rahmati et al. (2013) used multi-objective version of GA called NSGA-II and NREGA and also a MOHS algorithm to solve a multi-server location allocation problem where Taguchi was used to tune the algorithm's parameters. Due to novelty of the problem, there was no benchmark in the literature; hence they generated 20 different numerical examples to evaluate their algorithms. Pasandideh et al. (2013) solved a multi-objective multi-facility problem using GA and SA where a MODM approach was applied to combine the objectives into one objective. In their work, 20 different numerical problems were generated to evaluate the algorithms where the algorithms were compared in terms of CPU time and fitness function statistically and graphically. Rahmati et al. (2014) optimized a multi-objective queuing multi-facility problem using NSGA-II and NREGA where generated 15 different problems to evaluate the algorithms performance statistically and graphically.

There are a wide range of works in the literature used PSO (Akbari Kaasgari et al., 2017; Bhunia et al., 2017; Y. Cheng et al., 2015; Dabiri et al., 2017; Dye, 2012; Sadeghi et al., 2014b; Soleimani & Kannan, 2015; Srivastav & Agrawal, 2016; Tsai & Yeh, 2008; M. Yang & Lin, 2010; T. Zhang et al., 2015) and also GA

(Ahmadzadeh & Vahdani, 2017; Alikar et al., 2017; Azadeh et al., 2017; Çelebi, 2015; Hiassat et al., 2017; Kundu et al., 2017; B. Park et al., 2016; Roozbeh Nia et al., 2015; Saracoglu et al., 2014) to solve inventory-supply chain and location allocation and also lot-sizing problems.

2.4.2 FOA

In this research, a modified version of fruit fly optimization algorithm (MFOA) has been employed to solve a mixed binary-integer supply chain problem. This algorithm, which was first introduced by (Pan, 2012), is inspired by the behavior of fruit fly in finding the food. Recently, this algorithm has been taken into account by several authors. Zheng et al. (2013) used a novel fruit fly optimization to solve the semiconductor final testing scheduling problem. First, they presented a new encoding scheme to represent solutions reasonably. Second, they used a new decoding scheme to map solutions to feasible schedules. Finally, they utilized multiple fruit fly groups during the evolution process to enhance the parallel search ability of the algorithm. Zhang and Wang (2013) applied an improved fruit fly optimization algorithm to solve a lot-streaming flow-shop scheduling problem with equal-size sub-lots. In their proposed algorithm, a solution was represented as two vectors to determine the splitting of jobs and the sequence of the sub-lots simultaneously. Ling Wang et al. (2013) applied a novel binary fruit fly optimization algorithm to solve the multidimensional knapsack problem in which a binary string was used to represent the solution of the problem, and three main search processes were designed to perform evolutionary search, including smell-based search process, local vision-based search, and global vision-based search process. Li et al. (2013) proposed a hybrid annual power-load forecasting model combining fruit fly optimization algorithm and generalized regression neural network to solve their problem, where the fruit fly optimization was used to automatically select the

appropriate spread parameter value for the generalized regression neural network power-load forecasting model. Dai et al. (2014) found the necessity of an improvement where the smell-concentration judgment value was non-negative in (Pan, 2012). Ramachandran and Thomas Bellarmine (2014) presented a novel state estimator for minimizing the size of the phasor measurement unit (PMU) configuration while allowing full observability of the network. The proposed approach initially finds the best configuration of PMU for observability. They employed a novel meta-heuristic algorithm called improved fruit fly optimization to determine the minimum number of PMUs that could sustain observability. Table 2.2 contains a summary of the aforementioned relevant works and the place of the current work in comparison with them.

Ling Wang and Zheng (2013) solved a multi-objective version of a multi-skill resource-constrained scheduling problem using FOA to minimize the makespan and the total cost simultaneously. Mousavi et al. (2017) used a modified version of FOA mixed with fuzzy rule-based systems to optimize the classification accuracy of datasets in the literature at which algorithms' parameters were tuned using Taguchi method. A FOA was also improved by (Mousavi et al., 2016) to solve a series-parallel redundancy allocation problem under AUD and IQD policies where the results were compare to PSO, TS and GA algorithms. Taguchi method was designed to tune the parameters of the proposed algorithms. Zheng and Wang (2016) applied a two-stage knowledge-based FOA to solve unrelated parallel machine scheduling problem by some resource limitations where the aim was minimizing the makespan. They generated some numerical instances to evaluate the algorithm performance since there was no benchmark in the literature in which results were compared to another algorithm statistically, graphically and using CPU time. Shen et al. (2016) solved a support vector machine parameter tuning approach using FOA where GA,

PSO, BFO and Grid algorithms were applied to compare the FOA performance graphically, statistically and using CPU time. In (Lin Wang et al., 2015), an improved FOA was used to determine numerical functions and solving joint replenishment problems. A modified version of FOA also was applied by (Dai et al., 2015) to optimize the layout of IMUs in large ship for detecting the deformation of the deck. Yu et al. (2016) solved a multi-objective parameter of support vector regression to improve the generalization ability of machine learning in prediction purpose. A novel FOA was used by (Balasubbareddy, 2016) to solve single and multi-objective optimization problems with generation fuel cost, total power losses and voltage stability index as objectives. Wu et al. (2015) used a normal fuzzy cloud model to improve the performance of FOA as well. Cong et al. (2016) employed the FOA to improve the accuracy of the traffic flow forecasting in the field of modern intelligent transportation systems where the least squares support vector machine was presented along with the heuristic algorithms to find out the value of its two parameters.

2.4.3 Taguchi approach

It will be considerably time consuming to tune the parameters of the proposed metaheuristic algorithms manually while a combination of different values should be tested for each parameter one by one. The alternatives of using classical statistical optimization tools to set the control parameters, the orthogonal arrays of the Taguchi method (Ross, 1988) are usually applied to study more decision variables in smaller number of experiments.

It will take a long time from the algorithms to run them with different values of the parameters manually. In fact, when the proposed algorithms are run manually it needs to run for each value of the parameters several times individually which it will

last a long time to get the best values of the parameters with respect to combination of the different values. Determining the best initial parameter values for an algorithm, called parameter tuning, is crucial to obtaining better algorithm performance (Gümüş et al., 2016). In order to attain the best solutions in lesser time in the models formulated in this thesis, Taguchi approach was applied in this section to tune the control parameters of the proposed GA, PSO and FOA algorithms. In addition, to clarify that Taguchi performs very well especially in seasonal inventory, in Mousavi et al. (2014) Taguchi was compared with another common design of experiment method i.e. Response Surface Methodology (RSM) for both-single objective and multi-objective problems where Taguchi showed better performance than RSM in many different problems. In other words, the Taguchi method reduces the time spent on finding the combination of the optimal parameter values of the GA, PSO and FOA algorithms by running a smaller size of experiments on the training instances. Therefore, Taguchi is an efficient design of experiment method to tune the parameters of the metaheuristics in seasonal Inventory problem.

Finding the combinations of the algorithms' parameters traditionally without using Taguchi can highly effect on the performance of the proposed PSO, GA and FOA in terms of the run time and the accuracy that lead to find non-optimal parameters values and solutions.

Therefore, it will take the algorithms especially MOPSO and MOGA a long time to find the best combination of their parameters without using Taguchi when we should use trail and error approach. Furthermore, tuning the parameters of MOPSO and MOGA manually and without applying Taguchi will never guarantee to find the optimal values of their parameters and also the optimal solutions of the problem consequently.

Recently, many works in the literature have used Taguchi method to get the best values of the parameters of the algorithms applied in different problems. Azadeh et al. (2017) used Taguchi to tune GA algorithm, (Vahdani et al., 2017) applied Taguchi to tune GA and SA algorithms in a location-inventory problem. Rayat et al. (2017) employed Taguchi to set the SA, GA and PSO parameters in a multi-objective location-inventory. Mousavi et al. (2016) used Taguchi to tune the NSGA-II, MOPSO and NRGa parameters in a multi-objective inventory problem under inflation and time value of money. Ahmadzadeh and Vahdani (2017) utilized Taguchi to set the parameters of GA, ICA and FA algorithms in location-inventory and pricing problem. Sadeghi and Niaki (2015) used Taguchi to tune the parameters of NSGA and NRGa algorithms in fuzzy multi-objective inventory problem. Mousavi et al. (2014b) applied Taguchi method to set the PSO and HS algorithms' parameters in a fuzzy multi-objective inventory problem. Mohammadi (2015) and Mohammadi et al. (2015) applied Taguchi approach to set the parameters of GA, PSO, SA and ABC metaheuristic algorithms in a multi-facility lot-sizing inventory-supply chain problem. In works published in other fields also show how the Taguchi improves the metaheuristic algorithms performance (Alikar et al., 2017; Alikar et al., 2017; Sadeghi et al., 2013a). Rahmati et al. (2013) applied Taguchi approach to set the proposed algorithms parameters applied for solving a multi-facility location allocation problem. Fattahi et al. (2015) used Taguchi approach to set the parameters of five multi-objective metaheuristic algorithms as well as MOPSO, NSGA-II, NRGa, MOHS and MOVDO.

2.5 Summary

In this chapter, an overall literature review related to the proposed problem has been performed. The works accomplished in multi-item multi-period ICP were reviewed where some studies relevant to multi-objective version of multi-item and

multi-period ICP existed in the literature were given. Then, the ICP works done in fuzzy environment were considered. Moreover, the provided inventory-supply chain problem was introduced and reviewed where the studies multi-item and multi-period were investigated. The works related to supply chain, location allocation and supply chain-inventory problems for both cases of with and without shortages available in the literature were also investigated in this chapter. Furthermore, the surveys related to the proposed meta-heuristic algorithms performed in ICP and ISLAWOS and ISLAWS were reviewed. Some recent published works in lot-sizing problem were also reviewed. In addition, the works those used the metaheuristic algorithms as well as PSO, GA, SA and FOA were reviewed. Finally, some works applied Taguchi method for tuning the algorithms' parameters have been reviewed.

According to the review works performed in this chapter, little attention has been devoted to ICP, inventory-supply chain and location allocation problems. Hence, this research addresses these shortcomings by formulating comprehensive mathematical models and solution methods for such problems arising in reality.

Table 2.2: A brief review of the works related to ISLAWS in the literature

Authors and year	Single/Multi period	Single/Multi product	Discount	Space Constraints (Y/N)	Design of Experiment	Solving method	Joint supply chain and location (Y/N)	Shortages (Y/N)	Budget constraints (Y/N)
(Karakaya & Bakal, 2013)	Single	Multi	AUD	N	-	Numerical method	N	N	N
(Diabat, Richard, et al., 2013)	Single	Multi	-	N	-	Lagrangian relaxation-based	Y	N	N
(Diabat et al., 2009)	Single	Multi	-	N	-	Genetic algorithm	Y	N	N
(Ganesh et al., 2013)	Single	Single	-	N	-	Simulation analysis	N	N	N
(Tsao & Sheen, 2012)	Single	Multi	-	N	-	Heuristic	N	N	N
(Ramezani et al., 2014)	Multi	Multi	Fixed rate	Y	-	Fuzzy optimization	Y	N	N
(DZ Zhang, 2013)	Multi	Multi	-	N	-	Decision making	N	N	N
(Mousavi, Hajipour, Niaki, & Alikar, 2013)	Multi	Multi	AUD, IQD	Y	Taguchi	Genetic algorithm	N	N	Y
(Rodriguez et al., 2014)	Multi	Multi	-	N	-	Linearization	N	Y	N
(D, 2013)	Multi	Single	-	N	-	Game theory	N	N	N
(Lin, 2013)	Multi	Multi	Fixed rate	N	-	Heuristic	N	Y	N
(Tsao & Lu, 2012)	Single	Single	Quantity transportation	N	-	Non-linear technique	Y	N	N
(J. Li & Liu, 2006)	Multi	Single	IQD	N	-	Heuristic	N	Y	N
(Mousavi, Bahreininejad, et al., 2014)	Multi	Multi	AUD	Y	Taguchi	PSO	Y	N	Y

Table 2.2, Continued

(K.-L. Huang et al., 2014)	Multi	Single	-	N	-	Heuristic	N	N	N
(C.-T. Chang & Chou, 2013)	Multi	Single	-	N	-	Exact method	N	N	N
(Mirzapour Al-e-hashem et al., 2013)	Multi	Multi	IQD	N	-	Linear solver	N	Y	N
(Roozbeh Nia et al., 2013)	Multi	Single	-	Y	-	Ant colony algorithm	N	Y	N
(Taleizadeh et al., 2012)	Multi	Single	-	N	-	PSO	N	Y	Y
(Mousavi, Niaki, et al., 2013)	-	-	-	N	Taguchi	Genetic algorithm	N	N	N
(Mousavi & Niaki, 2013)	-	-	-	N	Taguchi	Genetic algorithm	N	N	N
The current work	Multi	Multi	AUD, IQD	Y	Taguchi	MFOA	Y	Y	Y

CHAPTER 3: METHODOLOGY

3.1 Introduction

This Chapter explains the proposed research methodology applied in this work. First, all the stages performed to provide the research are explained in Section 3.2 where a Flowchart is drawn to show the details step by step graphically. Then, the three problems proposed in this research are described in Sections 3.3 to 3.5 respectively. In Section 3.6 the solution methodologies are expressed in which four metaheuristic algorithms as well as GA, FOA, PSO and SA are employed to solve the problems. In addition, Taguchi method applied for tuning the parameters of the proposed algorithms is explained in Section 3.6. Finally, the summary of the chapter comes in Section 3.7.

3.2 Research Steps

At the first step of the methodology used in this research, a literature review on different aspects of inventory control and supply chain problems was performed. Furthermore, a variety of the problems in supply chain and location allocation fields was studied in order to identify the gaps and problems existing in the literature. Then, the particular problems were identified and classified and for each one of these problems suitable solution methods were determined. The proposed problems were classified into two types which are multi-objective and single-objective problems. At the first part of the research, a multi-objective multi-item multi-period inventory planning problem with total available budget under AUD for some items and IQD for other items was considered. The orders were assumed to be placed in batch sizes. Shortages were allowed and contain backorder and lost sale. Besides, the manager decided to build a new or extend the old warehouse for the company to store more items there. The objectives were to minimize both the total inventory costs and the

total required storage space, for which a fuzzy weighted combination was defined as the objective function. The aim of the study was to determine the optimal order quantity and shortage quantity of each product in each period such that the objective function is minimized and the constraints hold.

The second part of the research focuses on optimizing two models of inventory-supply chain network and location allocation (ISLA) for a multi-product multi-period inventory system with multiple buyers (retailers), multiple vendors (distributors) and warehouses with limited capacity owned by the vendors. The inventory replenishment starts at a certain time-period and finish at another time-period where the buyers purchase the products from the vendors during these interval periods. The vendors provide (produce) the various products to the buyers with variable demand rates under AUD policy since the production capacity of each vendor is restricted. The vendors satisfy the buyers' demands immediately in all the periods so that no shortages occur in the first model while shortages are allowed during the replenishment as an extension of the model and in case of shortage, a fraction of demand is considered backorder and a fraction lost sale. When the demands of the buyers are satisfied in a period, the products remained from the period in addition to the ordering quantities of the next period enter into the warehouses. The total available budget for purchasing the products and also the total vendors' warehouse space are constrained. Moreover, the distance between the buyers and the vendors in the model is assumed as Euclidean distance since the distance is considered to be Euclidean and Square Euclidean in the extended model. The main goal is to find optimal locations of vendors between the buyers and to determine the order quantities of the products ordered by the buyers from the vendors in different periods so that the total supply chain costs are minimized.

At the same time the modeling and classifying process of the problem is being done, some solution methods were identified and classified to solve the particular problems. While the proposed problems in this research which are multi-item multi-period inventory control and ISLA problems have been approved to be NP-hard, meta-heuristic algorithms were the most appropriate methods to optimize the provided problems. Then, the best meta-heuristic algorithms were chosen after testing a lot of algorithms on the problems under investigation. In the current research, MPSO, PSO, GA, SA and MFOA algorithms are selected to optimize inventory control and ISLA problems. The mathematical models proposed in this work have been coded in MATLAB programming software. Due to the novelty of the models, there has been no benchmark in the literature to compare the results of the provided algorithms as well. Therefore, a wide range of numerical examples was generated to evaluate the efficiency of the selected approaches. In addition, to make the algorithms more effective, Taguchi method was utilized to tune different parameters of the algorithms.

A flow chart that summarizes the overall process of this project is shown in Figure 3.1.

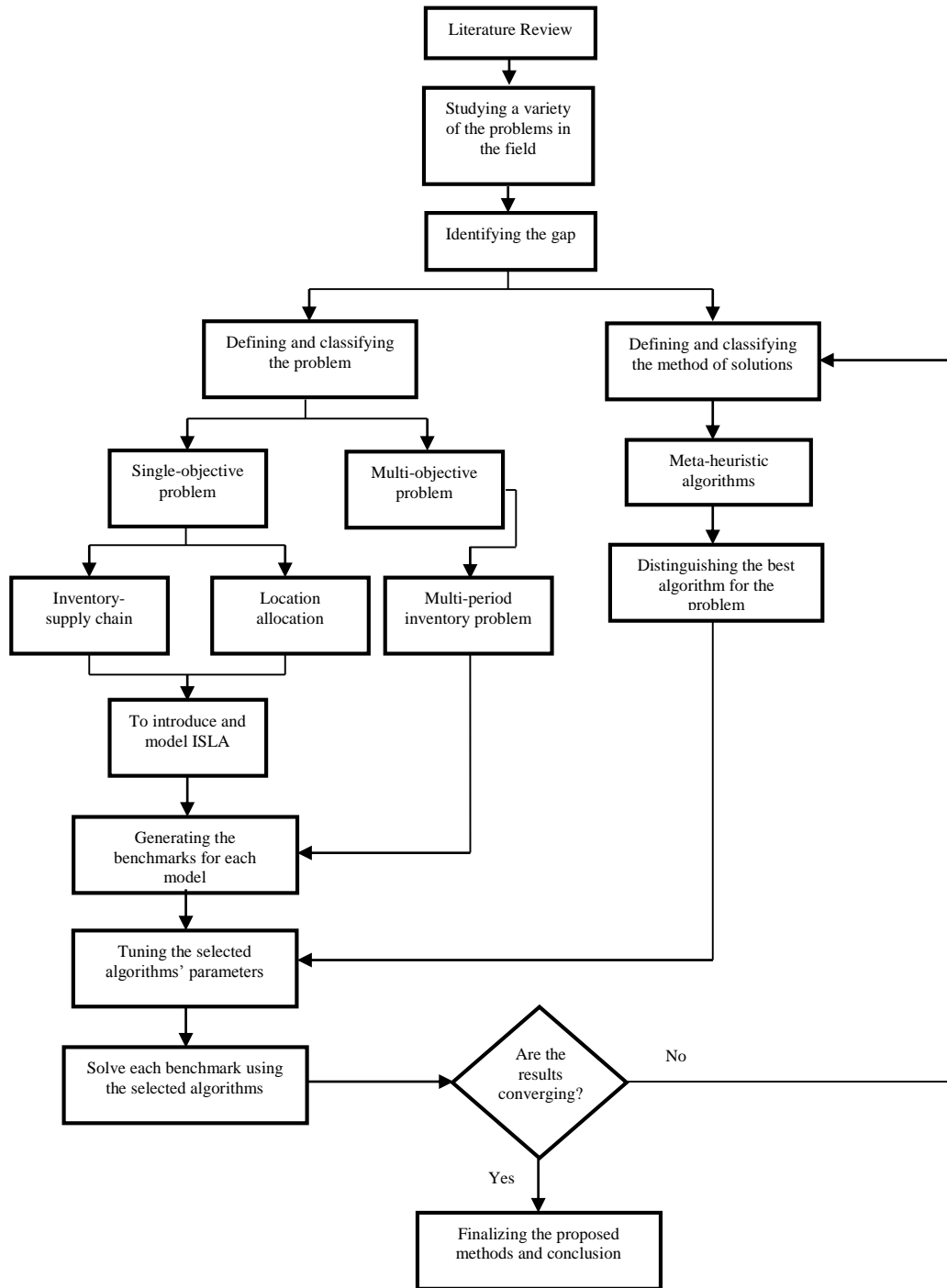


Figure 3.1: Methodological flowchart of this research

3.3 The proposed multi-item multi-period ICP

One of the models that has been applied extensively to formulate various inventory control problems is the EOQ. Many researches refer the history of EOQ to Harris (Abdel-Malek, Montanari, & Morales, 2004). This classic model

encompasses the planning for one product in a period with several assumptions. Although these assumptions make the model simple, the usability of the EOQ model in real-world situations is limited.

In this research, a mixed binary integer mathematical programming model for a multi-objective multi-item multi-periodic inventory control model was developed where some items were purchased under AUD and the other items were bought from IQD. The demands vary in different periods, the budget was limited, the orders were placed in batch sizes, and shortages in combination of backorder and lost sale were considered. The goal was to find the optimum order quantities of the items in each period such that the total inventory cost and the total required warehouse space were minimized simultaneously. Since it is not easy for the managers to allocate the crisp values to the weights of the objectives in a decision making process, considering these weights as fuzzy numbers will be taken as an advantage. Therefore, a FMODM approach i.e. FWSM was used to combine the objectives into one objective. In the multi-objective problems, all the objectives should be independent from each other and hence they are in conflict with each other where FWSM uses the weights to make all the objectives in one objective with the same unit (weight type).

In order to make the problem more understandable, the model was explained using the real world example. A company which produces some kinds of fashion clothes including trousers, t-shirt, and shirt in a certain period, was considered. The customers (wholesales) of this company with different demand rates make the orders and receive their products in the pre-specific boxes, each one consisting of a known number of these clothes. Moreover, due to some unforeseen matters, such as production limitation, the companies were not responsive to all of the demands in a period and hence some customers must wait until the next period to receive their orders. Furthermore, it was assumed that the company was going to extend the

production part and therefore the owner has a plan to build and optimize a new storage subject to the available space. As another example in the real world, a company that manufactured High Gloss and finishing items in Iran was considered.

While lot-sizing is defined as the quantity of an item demanded for delivery on a particular period (or multi-period) or manufactured in a single production run, the current study models a multi-item inventory problem for seasonal items with different production run lines which is different than lot-sizing problem in terms of formulation and complexity.

The description of the provided inventory control problem and employed algorithms are given in details in Chapter 4.

3.4 The proposed multi-item multi-period ISLAWOS

In the inventory control problems, the vendors sell their products under some discount policies in order to attract and encourage the customers to purchase more items. All unit discount (AUD) is one of the most common policies that has been taken into account in the literature, recently. The proposed problem aims to optimize a supply chain network for a multi-product multi-period inventory system with multiple buyers, multiple vendors and warehouses with limited capacity owned by the vendors. The inventory replenishment starts at a certain time-period and finish at another time-period where the buyers purchase the products from the vendors during these interval periods. The vendors provide (produce) the various products to the buyers with variable demand rates under all-unit discount policy since the production capacity of each vendor was restricted. The vendors satisfy the buyers' demands immediately in all the periods so that no shortages occur during the replenishment.

When the demands of the buyers were satisfied in a period, the products remained from the period in addition to the ordering quantities of the next period enter into the warehouses. The novelty of the problem is that the integrated supply chain expressed in this paper simultaneously determines two types of decision variables: (i) the locations of the vendors in a certain area among the buyers with fixed locations and (ii) the allocation: the order quantities of the products at each period made by the buyers from the vendors. The total available budget for purchasing the products and also the total vendors' warehouse space were constrained. Moreover, the distance between the buyers and vendors was assumed as Euclidean distance. The vendors store the products into their warehouses and then the products were transported from these warehouses to the buyers.

The description of the proposed ISLAWOS and the applied algorithms are given in details in Chapter 5.

3.5 The proposed multi-item multi-period ISLAWOS

Supply chain is an integrated system of facilities and activities that synchronizes inter-related business functions of material procurement, material transformation to intermediates and final products, and distribution of these products to customers. SCM is a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed at the right quantities, to the right locations, and at the right time, in order to minimize system-wide costs while satisfying service level requirements (Simchi-Levi et al., 2000).

In the provided model, a mixed binary-integer nonlinear mathematical model was developed for a location–allocation two-echelon retailer–distributor supply chain problem in which variety of the products were offered by the distributors to retailers. The distributors distribute the products and store them in their own warehouses with

limited capacities where there are certain places to hold each item. The retailers order different products to different distributors at specific time-periods based on their requirements. The products were delivered in certain packets with specific number of items using trucks with limited capacities. The planning horizon of the problem comprised multiple periods, where the replenishment process is taking place at the beginning of these periods. Here, different products in different periods may face shortages as a combination of lost sales and backorders. Moreover, it is possible the whole quantities of the products cannot be sold in different periods. Therefore, a number of items remain in the warehouses. In case of a shortage for a product in a period, the retailer should make an order at least as much as the demand becomes satisfied. Besides, due to some uncertain constraints, the distributors were not able to produce the products more than a specific value and also the total available budget was limited. The distributors provide the products to the retailers under all-unit and incremental quantity discount policies. The main goal was to find the optimal locations of distributors among the retailers and to determine the order quantities of the products ordered by the retailers to the distributors in different periods so that the total supply chain costs were optimized.

The description of the investigated ISLA and the used algorithms was explained in details in Chapter 6.

3.6 Solution methods

In recent decades, scientists have been mimicking natural phenomena to propose methods and algorithms for solving complex optimization problems. Based on the complexity of real-life optimization problems, one may not be able to use exact algorithms.

Therefore, typically, meta-heuristic methods are frequently used to find a near optimum solution in an acceptable period. Meta-heuristics are kind of near-optimal algorithms that were proposed in the two recent decades to integrate basic heuristic methods in higher-level structures in order to effectively and efficiently search a solution space. Nowadays, these algorithms have a large number of applications in optimization of different hard-to-solve problems.

This thesis formulates the mathematical models for three different problems, which are applicable in many real industries with a wide range of constraints as well as production limitations, space limitation for storing items, transportation limitations and also limitation in packing the items. Therefore, the proposed models with such amounts of limitations and mixed-integer binary nonlinear objective functions are complicated and it is impossible for them to use normal optimizer solvers to find the optimal solutions. While supply chain and inventory problems are approved to be strongly NP-hard, in this research several meta-heuristic algorithms were used to solve the problems where MOPSO and MOGA were applied to optimize the first proposed model, the modified PSO and GA for the second model and MFOA, PSO and SA for the third provided model. Based on the works reviewed in Chapter 2, these algorithms are found to be the most popular in the literature of the fields of Inventory and supply chain problems. The problems proposed here are formulated to be mixed-integer binary nonlinear programming with several real world constraints which cause the problem becomes really complicated. Solving the problems has been performed using standard solvers as well as CPLEX, which is the most reliable one in the literature, but the problems were not solved after running a few days in a row. Therefore, metaheuristics were the best choice to solve the problems while the proposed problems have been approved to be considerably NP-hard.

To justify the use of different solution methodologies (meta-heuristic algorithms) for the different problems proposed in this thesis, the reasons are the following. The first model formulated in the first problem proposed in Chapter 4 is a multi-objective extension of the work proposed by (Mousavi et al., 2013) in which a GA, a Branch & Bound (B&B) and also a SA were used to solve the problem. The GA was found to be the best in terms of total cost value and CPU time. Furthermore, GA was found to be one of the most common algorithms in the literature of inventory problem according to the works reviewed in Chapter 2. Therefore, a modified version of GA is used to be compared with a PSO algorithm to show the quality of the solutions of the algorithms. In this work, all the problems have been first coded by using a standard solver as well as CPLEX solver like the way done in (Mousavi et al., 2013) where the solver couldn't find the optimal solutions after running a few days in a row due to the highly complexity of the problems. The results of CPLEX in (Mousavi et al., 2013) also showed the weakness of the solver which was unable to solve problems with large size and it was run a long time without any outcome while GA reached the near optimal solutions in a very lesser CPU time. Therefore, GA was chosen for solving this problem while PSO and FOA also have been used as two other efficient metaheuristic algorithms according to the literature reviewed in Chapter 2. After running these algorithms on the proposed problem with 40 different generated numerical examples with different sizes, and also on the case study, the PSO has showed better performance than other algorithms in terms of inventory cost and CPU time, where the results of PSO and GA have been brought in Chapter 4. PSO, GA, and FOA have been also applied to solve the second and third models proposed in Chapters 5 and 6 in which GA and PSO have had better performance than FOA in terms of the total costs in Chapter 5. Against, the results have been in favor of FOA in comparison with other algorithms applied to solve the problem

proposed in Chapter 6. In order to show the quality of the solutions obtained by these algorithms in Chapter 6, FOA which is classified as population-based algorithm has been also compared to a neighborhood-based algorithm i.e. SA in addition to GA in terms of total costs and CPU time.

3.6.1 Particle Swarm Optimization

The particle swarm optimization (PSO) proposed by (Kennedy & Eberhart, 1995) is a population-based stochastic meta-heuristic algorithm that was inspired by social behavior of bird flocking or fish schooling. PSO is a meta-heuristic that requires few or no assumptions on the problem being optimized and can search very large spaces of candidate solutions. PSO can therefore be used on optimization problems that are partially irregular and noisy over time (Gigras & Gupta, 2012).

The overall flowchart of the PSO algorithm is shown in Figure 3.2. The PSO algorithm improved for the related model is provided in details in Chapters 4 and 5.

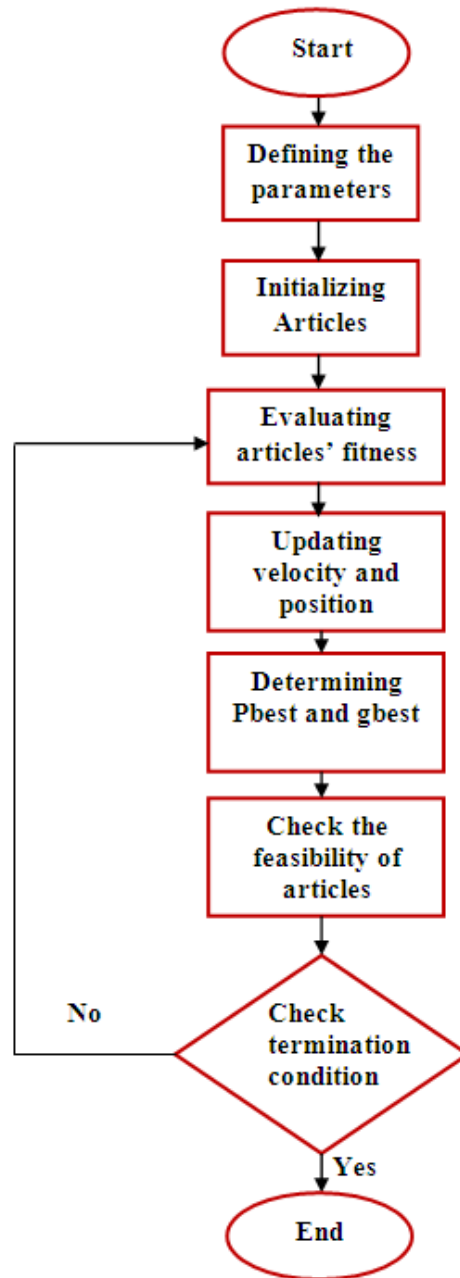


Figure 3.2: The flowchart of the proposed PSO algorithm

3.6.2 Genetic Algorithm

GA is similar to computational models inspired by evolution. This algorithm encodes a possible solution to a specific problem on a simple chromosome-like data structure and applies recombination operators to these structures to preserve critical information. GA approach was first developed by (Holland, 1992) using the name “genetic plan”, and it attracted considerable attention as a methodology for search,

optimization, and learning after Goldberg's (Holland, 1992) publication. The first step in the implementation of any genetic algorithm is to create an initial population. Each member of this population will be a string. After creating an initial population, each string is then evaluated and assigned a fitness value. One generation is broken down into a selection phase and recombination phase. To create the next generation, new chromosomes, called offspring, are formed by either crossover operator or mutation operator. In short, the steps involved in the proposed GA algorithm are as follows:

- I. Setting GA parameters including the crossover probability (P_c), the mutation probability (P_m), population size (PS), and number of generation (NG).
- II. Initializing the population with the size of PS randomly.
- III. Evaluating the objective function.
- IV. Selecting individual for mating pool by tournament selection method and using elitisms.
- V. Applying the crossover operation for each pair of chromosomes based on P_c .
- VI. Applying mutation operation for each chromosome based on P_m .
- VII. Replacing the current population by the resulting new population.
- VIII. If stopping criteria is met, then stop. Otherwise, go to Step III.

In order to clarify the steps involved in the GA implementation of this research, only the main features of the algorithm is discussed in more details. One of the most important features is the chromosome representation.

The flowchart of the proposed GA is shown by Figure 3.3 where is going to be applied in the first and second models described in Chapters 4 and 5, respectively.

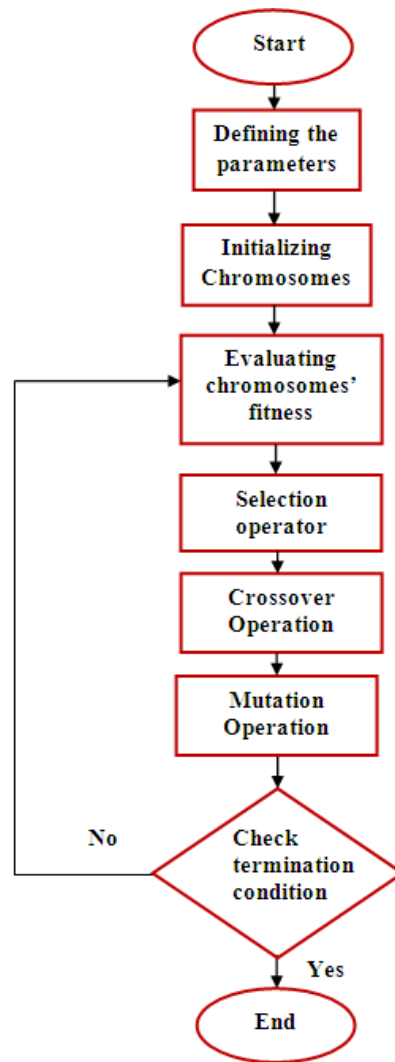


Figure 3.3: The flowchart of the proposed GA

3.6.3 Fruit Fly Optimization

The fruit fly optimization algorithm, first introduced by (Pan, 2012), is inspired by the food finding behavior of the fruit fly. The fruit fly itself was superior to other species in sensing and perception, especially in osphresis and vision. The osphresis organs of fruit flies can find all kinds of scents floating in the air; it can even smell food source from 40 km away. Then, after it gets close to the food location, it can also use its sensitive vision to find food and the company's flocking location, and fly towards that direction (Pan, 2012).

The flowchart of the modified version of the fruit fly algorithm called MFOA is given by Figure 3.4. The details of the provided fruit fly algorithm, applied to solve the third model, are described in Chapter 6.

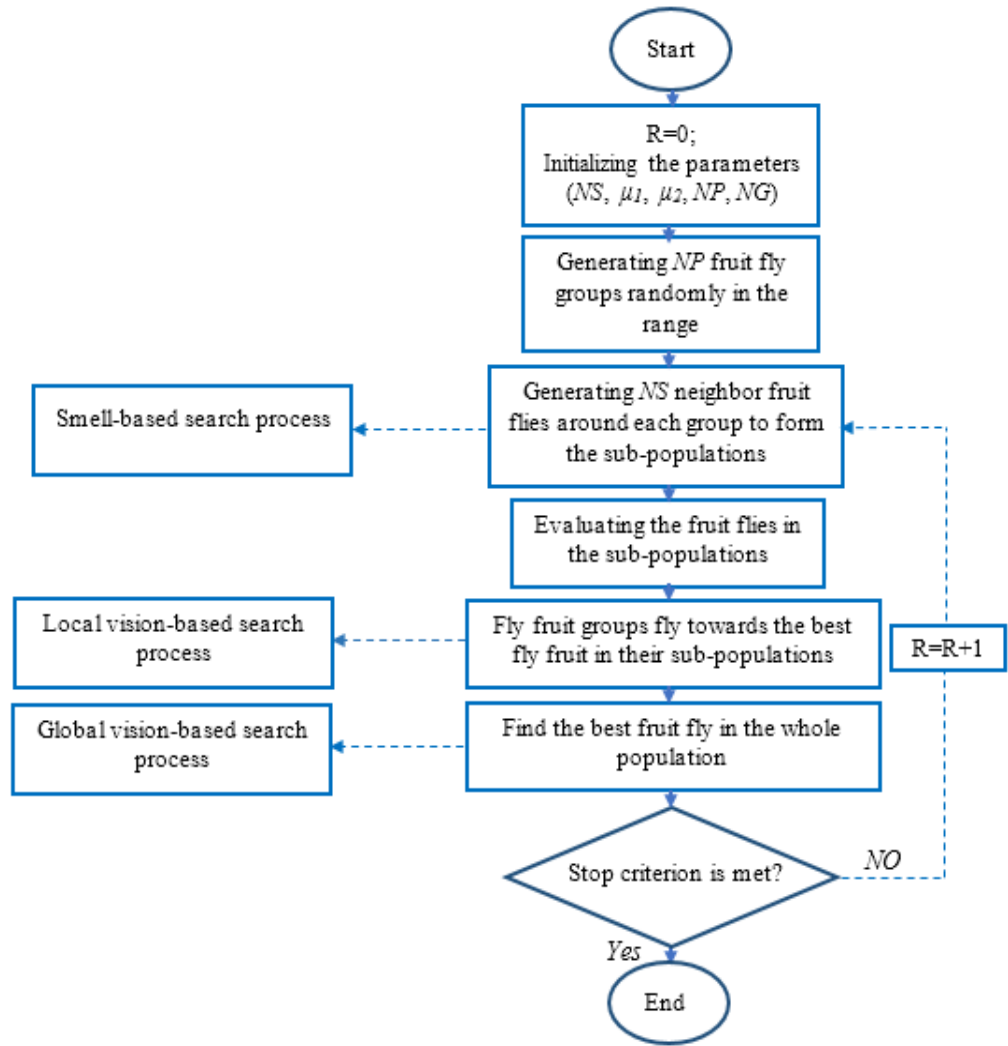


Figure 3.4: The flow chart of the proposed MFOA

3.6.4 Simulated Annealing

Simulated annealing (SA) is a well-known local search meta-heuristic algorithm introduced by (Kirkpatrick et al., 1983) who took the idea used to simulate physical annealing processes and developed it to solve complicated combinatorial optimization models. The SA of this research involves three control parameters b (the cooling schedule), A_0 (the initial temperature or the stopping rule of the external

loop), and *Gen* (the stopping rule of the internal loop). The solution representation and the evaluation in SA are similar to GA. Moreover, the mutation operator of GA is used as the neighborhood structure of SA as well. Figure 3.5 shows a pseudo-code of the proposed SA algorithm.

```

Select an initial Temperature  $T' > 0$ ;
Select an initial solution  $R_0$  and make it the current solution  $R$  and current best solution  $R^*$ 
;
Repeat
  Set repetition counter  $n = 1$ ;
  Repeat
    Generate solution  $R_n$  in the neighborhood of  $R$  ;
    Calculate  $\Delta = f(R_n) - f(R)$ ;
    If  $(\Delta \leq 0)$  then  $R = R_n$ ;
    Else  $R = R_n$  with the probability of  $L = e^{-\Delta/T'}$  where  $T' = T' \times \beta$ ;
    If  $(f(R_n) < f(R^*))$  then  $R^* = R_n$ ;
     $n = n + 1$ 
  Until  $n >$  number of repetitions allowed at each temperature level ( $E$ );
  Reduce the temperature  $T'$ ;
Until stopping criterion is true;

```

Figure 3.5: The pseudo-code of the proposed SA

3.6.5 Taguchi method

However, most researchers often fix parameters and operators manually based on the reference values of the previous similar studies. The Taguchi method reduces the time spent on finding a good parameter value combination by running a smaller size of experiments on the training instances from different domains as opposed to evaluating all combinations (Gümüş et al., 2016).

While classical statistical optimization tools are alternatives to tune the control parameters, the orthogonal arrays of the Taguchi method (Ross, 1988) are usually performed to study more decision variables in smaller number of experiments. Taguchi divides the factors into two main classes: controllable and noise factors. Noise factors are those that cannot be controlled directly. Since elimination of the noise factors is impractical and often impossible, the Taguchi method looks for minimizing the effect of noise while determining the optimal level of important controllable factors (Phadke, 1995). Taguchi created a transformation of the reiteration data to another value that is the measure of variation. The transformation is the signal-to noise (S/N) ratio. The term “signal” represents the desirable value (mean response variable) and “noise” represents the undesirable value (standard deviation). Thus, the S/N ratio implies the amount of variation present in the response variable (Phadke, 1995). Taguchi classifies objective functions into three groups: the “smaller the better”, the “larger the better”, and “the nominal is best” types. Since almost all objective functions in inventory control systems are grouped in the “smaller the better” type, its corresponding S/N ratio is:

$$S / N = -10 \text{Log}_{10}(\text{objective function})^2 \quad (3.1)$$

The Taguchi results of the optimal levels of the algorithms’ parameters tested on the three proposed models are given in Chapters 4, 5 and 6.

3.7 Summary

In this chapter, the methodology applied in the current work has been expressed and explained by a flowchart. The proposed research was divided into three main models, a multi-item multi-period inventory control problem, the multi-item multi-period ISLA where the shortage were not allowed, and a multi-item multi-period ISLA where shortages are allowed.

A brief overview of the mechanisms of a number of meta-heuristic approaches, namely, PSO, GA, SA, and MFOA was also described, which were used to deal with the intricacy of the proposed models and reach near-optimal to optimal solutions in a reasonable computation time.

CHAPTER 4: MULTI-ITEM, MULTI-PERIODIC INVENTORY CONTROL PROBLEM WITH VARIABLE DEMAND AND DISCOUNTS: A PARTICLE SWARM OPTIMIZATION ALGORITHM

4.1 Introduction

This chapter investigates a novel bi-objective multi-item multi-periodic inventory control model where some items were purchased under AUD and the other items were bought IQD. The demands vary in different periods, the budget was limited, the orders were placed in batch sizes, and shortages in combination of backorder and lost sale were considered. The goal was to find the optimum order quantities of the items in each period such that the total inventory cost and the total required warehouse space were minimized simultaneously. Since it is not easy for the managers to allocate the crisp values to the weights of the objectives in a decision making process, considering these weights as fuzzy numbers will be taken an advantage.

In order to understand the problem more, the model was explained using a real world example. A company, which produces some kinds of fashion clothes including trousers, t-shirt and shirt in a certain period, is assumed. The customers (wholesales) of this company with different demand rates make the orders and receive their products in the pre-specific boxes, each one consisting of a known number of these clothes. Moreover, due to some unforeseen matters such as production limitation, the companies were not responsive to all of the demands in a period and hence some customers must wait until the next period to receive their orders. Furthermore, it was assumed the company was going to extend the production part and therefore the owner has a plan to build and optimize a new storage subject to the available space.

The remainder of the chapter is organized as follows. In Section 4.2, the problem along with its assumptions is defined. In Section 4.3, the defined problem of Section 4.2 was modeled. In order to do this, the parameters and the variables of the problem were first introduced. A MOPSO algorithm was presented in Section 4.4 to solve the model. In section 4.5, a MOGA is also represented for solving the problem as a benchmark for comparisons. Section 4.6 proposes the numerical example containing both the generated problems and a case study. Finally, conclusion and recommendations for future research are presented in Section 4.7.

4.2 Problem definition, assumptions, and notations

In the real world problems due to competitive environment in the market, companies prefer to produce different products to satisfy all the needs of the customers. Besides, these companies plan to manufacture and propose their products for a pre-specific time-period (season) and then again try to compete with new arrival items in another specific time period. This strategy causes the customers always look for new arrival items with new designs in each period (season) which highly affects attracting more customers. For a company planning to store and sell its products in a certain area, it is really essential that how many Square feet or meter of that area should be allocated to the storage so that all the items can be stored in the least storage area.

Due to lack of study in the field in the literature, this chapter investigates a model for these sorts of companies to formulate a multi-objective multi-period (seasonal) inventory control problem at which the total inventory costs of the items and the total required storage space to store these items are minimized, simultaneously.

The main focus of this study is on the inventory problem with multiple items in multi-period (seasonal or in finite horizon). In the problem proposed in Chapter 4, a

novel inventory model for those companies producing multiple items is formulated to satisfy the demands of the customers during the different periods (seasons). The demand values proposed by the marketing team of the companies are variable in the different periods.

In the model considered in this study, the items are ordered based on the demand values estimated in each period or season where the prices of the items may vary in each period than another. Due to the fluctuations of the order quantities of these items during the time, companies should manage the delivering and storing processes of these items. Therefore, it is really essential for the companies to find out how much least space should be allocated to build a storage so that all the items can be placed in that in addition to finding out the minimum of the total costs.

The chapter investigates a model for these sorts of companies to formulate a multi-objective multi-period inventory control problem at which the total inventory costs of the items and the total required storage space to store these items are minimized, simultaneously. A multi-objective decision making approach called Weighted Sum Method which is the most common method in the literature is used to set the objectives into a single objective by multiplying each objective with a user supplied weight. The weight values of objectives completely depends on the objective type and are uncertain. In this study, in order to overcome this uncertainty the weights are considered as fuzzy variables. The content of this chapter has been published in (Mousavi et al., 2014; Mousavi et al., 2014a). There are also many works in the literature used Fuzzy Weighted Sum Method (H. Cheng et al., 2013; Ma et al., 2017; Majidi, Nojavan et al., 2017; Stojiljković, 2017; Su, 2017), in Supply chain and Inventory (Lee, 2017; Maity & Maiti, 2008) .

Consider a bi-objective multi-item multi-period inventory control problem, in which an AUD policy was used for some items and an IQD policy for some other

items. The inventory control problem of this research is similar to the items producing in different periods where the planning horizon starts in a period (or season) and finish in a certain period (or season). The total available budget in the planning horizon is limited and fixed. Due to existing ordering limitations or production constraints, the order quantities of all items in different periods cannot be more than their predetermined upper bounds. The demands of the products are constant and distinct, and in case of shortage, a fraction is considered backorder and a fraction lost sale. The costs associated with the inventory control system are holding, backorder, lost sale, and purchasing costs. Moreover, due to current managerial decision adaptations on production policies (i.e. building a new storage area), minimizing the total storage space was required as well as minimizing the total inventory costs. Therefore, the goal was to identify the optimal order quantities of the items in each period such that the two objective functions, total inventory costs and total storage space, were minimized.

The model formulated in this chapter is an extension of the study proposed in (Mousavi et al., 2013) while this model has been improved in several parts efficiently as some were mentioned in Table 2.1. In (Mousavi et al., 2013) the objective was to minimize total inventory costs while in this chapter the objectives are to minimize the total inventory costs and also to minimize the total required storage space built for storing the items. Furthermore, in the current model shortages are allowed in combinations of backorder and lost sale which are considered as an extra costs in the model while in (Mousavi et al., 2013) shortages weren't allowed. The costs were calculated under inflation and time values of money in (Mousavi et al., 2013) since in the current model the costs are computed without considering inflation and time values of money. Therefore, all formulas of the total costs will be different in (Mousavi et al., 2013) than the formula here. In addition, the truck

capacity for transferring the items are limited here while this constraint was not considered in (Mousavi et al., 2013). Moreover, in this chapter a FMODM i.e. FWSM is applied to combine the objectives into one function while a triangular fuzzy approach, a α -cut method and centroid defuzzification procedure are also used to show the fuzzy numbers and convert the fuzzy numbers to crisp values respectively. Additionally, in the work presented by (Mousavi et al., 2013) three GA, SA and B&B algorithms were used to solve 15 different problems while in this model multi-objective version of PSO and GA are employed for solving 40 different problems. Finally, a real case study has been studied in a wood industry in this work whilst no case study was performed in (Mousavi et al., 2013). By the way, some of these extensions were recommended in the Conclusion section of our previous work (Mousavi et al., 2013) (See Conclusion and future research work section of (Mousavi et al., 2013)). In addition, in the models proposed in Chapters 5 and 6, a novel supply chain problems and location-allocation problems of the model investigated by (Mousavi et al., 2013) have been extended.

In order to simplify the modeling, the following assumptions are set to the problem at hand.

- i. The demand rate of an item was independent of the others and was constant in a period. However, it can be different in different periods.
- ii. At most, one order can be placed in a period. This order can include or exclude an item.
- iii. The items were delivered in a special container. Thus, the order quantities must be a multiple of a fixed-sized batch.
- iv. The vendor uses an AUD policy for some items and an IQD policy for others.
- v. A fraction of the shortages is considered backorder and a fraction lost sale.
- vi. The initial inventory order quantity of all items was zero.

- vii. The budget was limited.
- viii. The planning horizon was finite and known. In the planning horizon, there were N periods of equal duration.
- ix. The order quantity on an item in a period is greater than or equal to its shortage quantity in the previous period (which are, $Q_{i,j+1} \geq b_{i,j}$ defined below.)

Figure 4.1 shows a graphical representation of the problem.

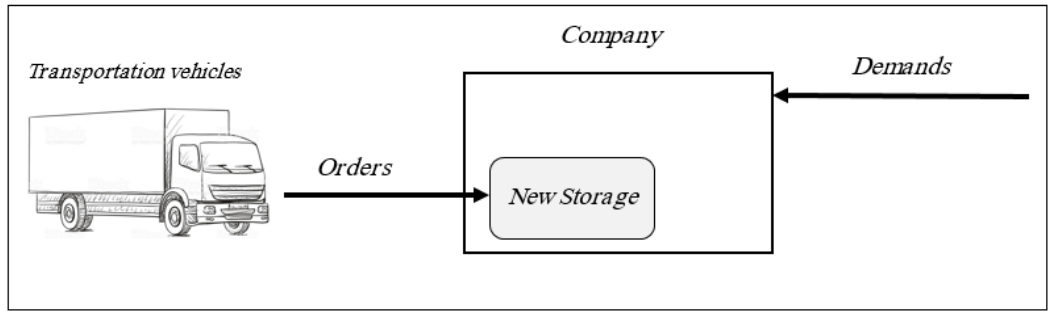


Figure 4.1: A graphical illustration of the proposed inventory control problem

In Figure 4.1, first the customers' demands of the items are received and then the company will order the items based on these demands. The owner of company tries to find the optimal required space of a new storage according to the optimal order quantity. The items will reach the company by trucks with the specific capacity. In order to model the problem at hand, in what comes next we first define the variables and the parameters. Then, the problem is formulated in Section 3.

The indexes, the parameters and the decision variables of the model are defined as follows:

Indexes:

$i = 1, 2, \dots, m$ is the index of the items

$j = 1, 2, \dots, N$ is the index of the periods

$k = 1, 2, \dots, K$ is the index of the price-break points

Parameters:

N : Number of replenishment cycles during the planning horizon

m : Number of items

K : Number of price break-points

S_i : Required storage space per unit of the i -th product

T_j : Total time elapsed up to and including the j -th replenishment cycle

$T'_{i,j}$: j -th period in which the order quantity of item i is zero (a decision variable)

B_i : Batch size of the i -th product

$D_{i,j}$: Demand of the i -th product in period j

A_i : Ordering cost per replenishment of product i (If an order is placed for one or more items in period j , this cost appears in that period)

$I_{i,j}$: Inventory position of the i -th product in period j (it is $X_{i,j+1} + Q_{i,j+1}$, if $I_{i,j} \geq 0$, otherwise equals $b_{i,j}$)

$I_i(t)$: The order quantity of the i -th item at time t

H_i : Unit inventory holding cost for item i

$q_{i,k}$: k -th discount point for the i -th product ($q_{i,1} = 0$)

$m_{i,k}$: Discount rate of item i in k -th price break-point ($m_{i,1} = 0$)

P_i : Purchasing cost per unit of the i -th product

$P_{i,k}$: Purchasing cost per unit of the i -th product at the k -th price break-point

$U_{i,j,k}$: A binary variable, set 1 if item i is purchased at price break-point k in period j , and 0 otherwise

$W_{i,j}$: A binary variable, set 1 if a purchase of item i is made in period j , and 0 otherwise

$L_{i,j}$: A binary variable, set 1 if a shortage for item i occurs in period j , and 0 otherwise

β_i : Percentage of unsatisfied demands of the i -th product that is back ordered

$\pi_{i,j}$: Back-order cost per unit demand of the i -th product in period j

$\hat{\pi}_{i,j}$: Shortage cost per unit of the i -th product in period j that is lost

Z_1 : Total inventory cost

Z_2 : Total storage space

TB : Total available budget

M_1 : An upper bound for order quantity of the i -th item in period j

M_2 : An upper bound for order quantities of all items in each period (the truck capacity)

TMF : Total multi-objective function value, which is the weighted combination of the total inventory cost and the total storage space

\tilde{w}_1 : A fuzzy weight associated with the total inventory cost ($0 \leq w_1 \leq 1$)

\tilde{w}_2 : A fuzzy weight associated with the total storage space ($0 \leq w_2 \leq 1$)

Decision variables:

$Q_{i,j}$: Purchase quantity of item i in period j (a decision variable where $Q_{i,j} = B_i V_{i,j}$)

$b_{i,j}$: Shortage quantity of the i -th product in period j (a decision variable)

$X_{i,j}$: The beginning positive order quantity of the i -th product in period j (in $j = 1$, the beginning positive order quantity of all items is zero), (a decision variable)

$V_{i,j}$: Number of the packets for the i -th product order in period j (a decision variable)

4.3 Problem formulation

A graphical representation of the inventory control problem at hand with 5 periods for item i was given in Figure 4.2 to obtain the inventory costs. At the beginning of the primary period (T_0), it was assumed that the starting order quantity of item i is zero and that the order quantity has been received and available. In the following periods, shortages can occur or not. If shortage occurs, the corresponding binary variable is 1, otherwise it is zero. In the latter case, the order quantities at the beginning of each period may be positive (Mousavi et al., 2014; Mousavi et al., 2013; Mousavi et al., 2016).

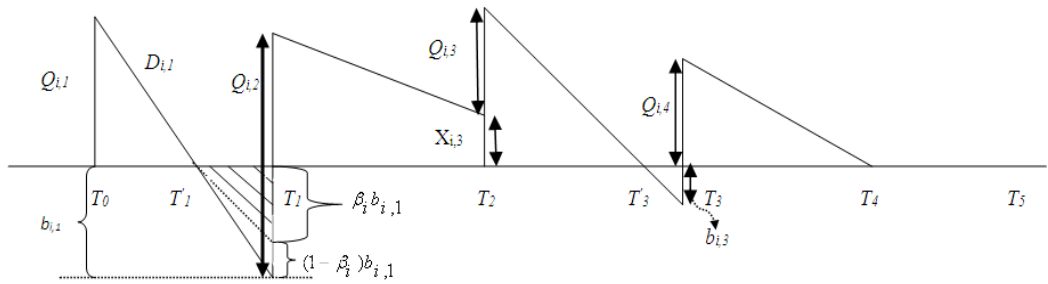


Figure 4.2: Some possible situations for the inventory of item i in 5 periods

4.3.1 The objective functions

The first objective function of the problem, the total inventory cost, is obtained as:

$$Z_1 = \text{Total Inventory Cost} = \text{Total Ordering Cost} + \text{Total Holding Cost} + \text{Total Shortage Cost} + \text{Total Purchasing Cost} \quad (4.1)$$

where each part is derived as follows.

The ordering cost of an item in a period occurs when an order is placed for it in that period. Using a binary variable $W_{i,j}$, where it is 1 if an order for the i -th product in period j is placed, and zero otherwise, and knowing that orders can be placed in periods 1 to $N-1$ (no order is placed in the period N) the total ordering cost is obtained as:

$$\text{Total Ordering Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} A_i W_{i,j} \quad (4.2)$$

Since it is assumed a shortage may occur for a product in a period or not, the holding cost derivation is not as straightforward as the ordering cost derivation. Taking advantages of a binary variable $L_{i,j}$, where it is 1 if a shortages for item i in period j occurs and otherwise zero, and using Figure 1, the holding cost for item i in the time interval $T_{j-1} \leq t \leq T_j(1-L_{i,j}) + T'_{i,j}L_{i,j}$ is obtained as:

$$H_i \int_{T_{j-1}}^{T_j(1-L_{i,j}) + T'_{i,j}L_{i,j}} I_i(t) dt \quad (4.3)$$

where $I_i(t)$ is the order quantity of the i -th item at time t .

In Equation (4.3), if a shortage for item i occurs, $L_{i,j}$ becomes 1 and the term $T_j(1-L_{i,j}) + T'_{i,j}L_{i,j}$ becomes $T'_{i,j}$. Otherwise, $L_{i,j} = 0$ and $T_j(1-L_{i,j}) + T'_{i,j}L_{i,j} = T_j$. In Figure 4.1, the trapezoidal area above the horizontal time line in each period when multiplied by the unit inventory holding cost of an item, H_i , represents the holding cost of the item in that period. In other word, since

$$X_{i,j+1} = Q_{i,j} + X_{i,j}(1-L_{i,j}) - D_{i,j}(T_j - T_{j-1}) - b_{i,j}L_{i,j} \quad (4.4)$$

The Equation (4.3) becomes:

$$H_i \int_{T_{j-1}}^{T_j(1-L_{i,j})+T'_{i,j}L_{i,j}} I_i(t) dt = \frac{X_{i,j} + Q_{i,j} - D_{i,j}}{2} (T_j(1-L_{i,j}) + T'_{i,j}L_{i,j} - T_{j-1}) H_i \quad (4.5)$$

Therefore, the total holding cost is obtained in Equation (4.6).

$$Total \text{ Holding Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{X_{i,j} + Q_{i,j} + X_{i,j+1}}{2} \right) (T_j(1-L_{i,j}) + T'_{i,j}L_{i,j} - T_{j-1}) H_i \quad (4.6)$$

The total shortage cost consists of two parts; the total backorder cost and the total lost sale cost. In Figure 4.2, the trapezoidal area underneath the horizontal time line in each period (shown for the primary period) when multiplied by the back-order cost per unit demand of the i -th product in period j , $\pi_{i,j}$, is equal to the backorder cost of the item in that period. Therefore, the total backorder cost will be:

$$Total \text{ Backorder Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\pi_{i,j} b_{i,j+1}}{2} (T_j - T'_{i,j}) \beta_i \right) \quad (4.7)$$

Furthermore, since $(1-\beta_i)$ represents the percentage demands of the i -th product that is lost sale, the total lost sale becomes:

$$Total \text{ Lost Sale Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\hat{\pi}_{i,j} b_{i,j+1}}{2} (T_j - T'_{i,j}) (1-\beta_i) \right) \quad (4.8)$$

where, $b_{i,j+1} = D_{i,j} (T_j - T'_{i,j})$ ($b_{i,1} = 0$).

In other words, if the inventory level of product i in period j is positive (i.e. $I_{i,j} > 0$) we have $X_{i,j} > 0$ and $b_{i,j} < 0$, otherwise $X_{i,j} \leq 0$ and $b_{i,j} \geq 0$.

The total purchase cost also consists of two AUD and IQD costs. The purchasing offered by AUD policy is modeled by

$$P_i = \begin{cases} P_{i,1} & ; & 0 < Q_{i,j} \leq q_{i,2} \\ P_{i,2} & ; & q_{i,2} < Q_{i,j} \leq q_{i,3} \\ \vdots & \\ P_{i,K} & ; & q_{i,K} < Q_{i,j} \end{cases} \quad (4.9)$$

Hence, the purchasing cost of this policy is obtained as:

$$AUD \text{ Purchasing Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} \sum_{k=1}^K P_{i,k} Q_{i,j} U_{i,j,k} \quad (4.10)$$

A graphical representation of the AUD policy employed to purchase the products in different periods is shown in Figure 4.3. In this Figure, the relation between the price break-points and the purchasing costs is demonstrated clearly. Moreover, $U_{i,j,k}$ is a binary variable, set 1 if the i -th item is purchased with price break k in period j , and 0 otherwise.

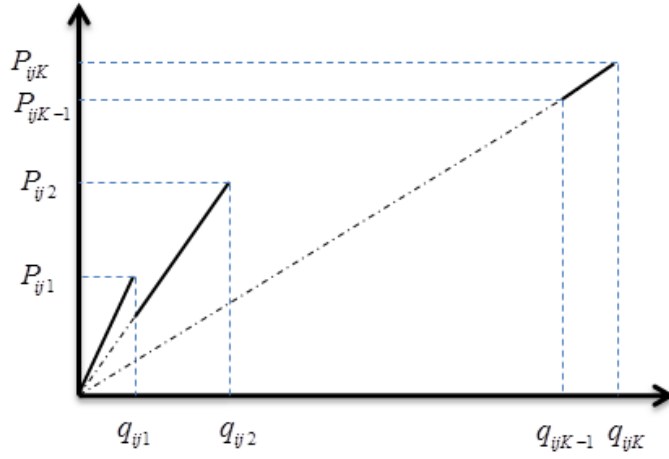


Figure 4.3: AUD policy for purchasing the products in different periods

In the IQD policy, the purchasing cost per unit of the i -th product depends on its order quantity. Therefore, for each price break-point we have:

$$\begin{cases} A_1 = P_{i,1} Q_{i,j} & ; 0 < Q_{i,j} \leq q_{i,2} \\ A_2 = P_{i,1} q_{i,2} + P_{i,2} (Q_{i,j} - q_{i,2}) (1 - m_{i,2}) & ; q_{i,2} < Q_{i,j} \leq q_{i,3} \\ \vdots & \vdots \\ A_3 = P_{i,1} q_{i,2} + P_{i,2} (q_{i,3} - q_{i,2}) (1 - m_{i,2}) + \dots + P_{i,K} (Q_{i,j} - q_{i,K}) (1 - m_{i,K}) & ; q_{i,K} < Q_{i,j} \end{cases} \quad (4.11)$$

Hence, the total purchasing cost under the IQD policy is obtained as:

$$IQD \text{ Purchasing Cost} = \sum_{i=1}^m \sum_{j=1}^{N-1} \left\{ (Q_{i,j} - q_{i,K}) P_i U_{i,j,K} (1 - m_{i,K}) + \sum_{k=1}^{K-1} (q_{i,k+1} - q_{i,k}) P_i (1 - m_{i,k}) \right\} \quad (4.12)$$

Figure 4.4 graphically depicts the IQD policy for each product in different periods in which A_1 , A_2 and A_3 are calculated in Equation (4.11).

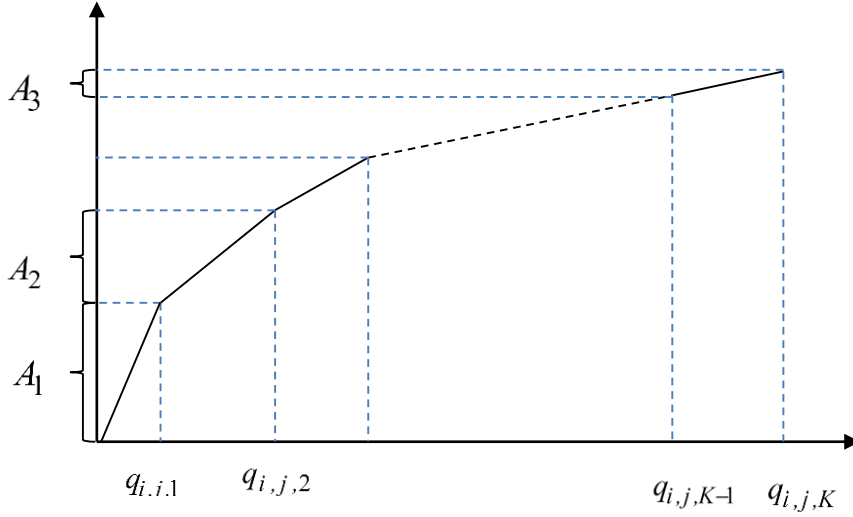


Figure 4.4: IQD policy for purchasing the products in different periods

Thus, the first objective function of the problem at hand becomes:

$$\begin{aligned} Z_1 = & \sum_{i=1}^m \sum_{j=1}^{N-1} A_i W_{i,j} + \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{X_{i,j} + Q_{i,j} + X_{i,j+1}}{2} \right) (T_j (1 - L_{i,j}) + T'_{i,j} L_{i,j} - T_{j-1}) H_i + \\ & \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\pi_{i,j} b_{i,j}}{2} (T_j - T'_{i,j}) \beta_i \right) + \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\hat{\pi}_{i,j} b_{i,j}}{2} (T_j - T'_{i,j}) (1 - \beta_i) \right) + \sum_{i=1}^m \sum_{j=1}^{N-1} \sum_{k=1}^K Q_{i,j,k} P_i U_{i,j,k} + \\ & \sum_{i=1}^m \sum_{j=1}^{N-1} \left\{ (Q_{i,j} - q_{i,K}) P_i U_{i,j,K} (1 - m_{i,K}) + \sum_{k=1}^{K-1} (q_{i,k+1} - q_{i,k}) P_i (1 - m_{i,k}) \right\} \end{aligned} \quad (4.13)$$

The second objective of the problem is to minimize the total required storage space. Since in each period, order quantities $Q_{i,j}$ enter the storage, and that the

beginning inventory of a period is the remained inventory of the previous period,

$X_{i,j}$, the second objective function of the problem is modeled by:

$$Z_2 = \sum_{i=1}^m \sum_{j=1}^{N-1} (X_{i,j} + Q_{i,j}) S_i \quad (4.14)$$

Finally, the fitness function is defined as the weighted combination of the total inventory cost and the required storage space using FWSM as

$$TMF = \tilde{w}_1 Z_1 + \tilde{w}_2 Z_2 \quad (4.15)$$

In the multi-objective problems, all the objectives must be independent from each other with different types (units) where the objectives are in conflict to each other. Therefore, the proposed FWSM is applied to make a single objective problem with the same unit or type (weight).

4.3.2 The constraints

In real world inventory planning problems, due to existing constraints on either supplying or producing goods (e.g., budget, labor, production and carrying equipment and the like), objectives are not met simple. This section presents formulations for some real-world constraints.

The first limitation is given in Equation (4.4), where it relates the beginning inventory of the items in a period equal to the beginning inventory of the items in the previous period plus the order quantity of the previous period minus the demand of the previous period.

The second limitation is due to delivering the items in packets of batches. Since $Q_{i,j}$ represents the purchase quantity of item i in period j , denoting the batch size by

B_i and the number of packets by $V_{i,j}$, we have:

$$Q_{i,j} = B_i V_{i,j} \quad (4.16)$$

Furthermore, since $Q_{i,j}$ can only be purchased based on one price break-point, the following constraint must hold:

$$\sum_{k=1}^K U_{i,j,k} = 1 \quad (4.17)$$

The prerequisite of using this strategy is that the lowest $q_{i,k}$ in the AUD table must be zero (which are $q_{i,1} = 0$).

Since the total available budget is TB , the unit purchasing cost of the product is P_i , and the order quantity is $Q_{i,j}$, the budget constraint will be

$$\sum_{i=1}^m \sum_{j=1}^{N-1} Q_{i,j} P_i \leq TB \quad (4.18)$$

In real world environments, the order quantity of an item in a period may be limited. Defining M_1 an upper bound for this quantity, for $i=1, 2, \dots, m$ and $j=1, 2, \dots, N-1$ we have

$$Q_{i,j} \leq M_1 \quad (4.19)$$

Moreover, due to transportation contract and the truck capacity, the number of product orders and the total order quantities in a period are limited as well. Hence, for $j=1, 2, \dots, N-1$, we have

$$\sum_{i=1}^m Q_{i,j} W_{i,j} \leq M_2 \quad (4.20)$$

where, if an order occurs for item i in period j , $W_{i,j} = 1$, otherwise $W_{i,j} = 0$. Further, M_2 is an upper bound on the total number of orders and the total order quantities in a period.

As a result, the complete mathematical model of the problem using FWSM is:

$$\text{Min } TMF = \tilde{w}_1 Z_1 + \tilde{w}_2 Z_2$$

Subject to:

$$\begin{aligned} Z_1 = & \sum_{i=1}^m \sum_{j=1}^{N-1} A_i W_{i,j} + \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{X_{i,j} + Q_{i,j} + X_{i,j+1}}{2} \right) (T_j (1 - L_{i,j}) + T'_{i,j} L_{i,j} - T_{j-1}) H_i + \\ & \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\pi_{i,j} b_{i,j}}{2} (T_j - T'_{i,j}) \beta_i \right) + \sum_{i=1}^m \sum_{j=1}^{N-1} \left(\frac{\hat{\pi}_{i,j} b_{i,j}}{2} (T_j - T'_{i,j}) (1 - \beta_i) \right) + \sum_{i=1}^m \sum_{j=1}^{N-1} \sum_{k=1}^K Q_{i,j} P_{i,k} U_{i,j,k} + \\ & \sum_{i=1}^m \sum_{j=1}^{N-1} \left\{ (Q_{i,j} - q_{i,K}) P_i U_{i,j,K} (1 - m_{i,K}) + \sum_{k=1}^{K-1} (q_{i,k+1} - q_{i,k}) P_i (1 - m_{i,k}) \right\} \end{aligned}$$

$$Z_2 = \sum_{i=1}^m \sum_{j=1}^{N-1} (X_{i,j} + Q_{i,j}) S_i$$

$$X_{i,j+1} = Q_{i,j} + X_{i,j} (1 - L_{i,j}) - D_{i,j} (T_j - T_{j-1}) - b_{i,j} L_{i,j};$$

$$(i = 1, 2, \dots, m) \text{ and } (j = 1, 2, \dots, N - 1)$$

$$b_{i,j+1} = D_{i,j} (T_j - T'_{i,j})$$

$$Q_{i,j} = B_i V_{i,j} \quad ; \quad (i = 1, 2, \dots, m) \text{ and } (j = 1, 2, \dots, N - 1)$$

$$\sum_{k=1}^K U_{i,j,k} = 1 \quad ; \quad (i = 1, 2, \dots, m) \text{ and } (j = 1, 2, \dots, N - 1)$$

$$\sum_{i=1}^m \sum_{j=1}^{N-1} Q_{i,j} P_i \leq TB$$

$$Q_{i,j} \leq M_1 \quad ; \quad (i = 1, 2, \dots, m) \text{ and } (j = 1, 2, \dots, N - 1)$$

$$\begin{cases} W_{i,j} \in \{0, 1\} \\ U_{i,j,k} \in \{0, 1\} \end{cases} \quad ; \quad (j = 1, 2, \dots, N - 1) \quad ;$$

$$(i = 1, 2, \dots, m), (j = 1, 2, \dots, N - 1), \text{ and } (k = 1, 2, \dots, K)$$

$$\sum_{i=1}^m Q_{i,j} W_{i,j} \leq M_2 \quad ; \quad (j = 1, 2, \dots, N - 1)$$

$$Q_{i,j+1} \geq b_{i,j} \tag{4.21}$$

In more inventory-planning models that have been developed so far, researchers have imposed some unrealistic assumptions such that the objective function of the model becomes concave and the model can easily be solved by some mathematical approaches like the Lagrangian or the derivative methods. However, since the model in Equation (4.21), which was obtained based on assumptions that were more realistic, is an integer nonlinear programming mixed with binary variables, reaching an analytical solution (if any) to the problem is difficult. In addition, efficient treatment of integer nonlinear optimization is one of the most difficult problems in practical optimization (Lee & Kang, 2008). As a result, in the next section a meta-heuristic algorithm is proposed to solve the model formulated in Equation (4.21).

4.4 The proposed MOPSO

Many researchers have successfully used meta-heuristic methods to solve complicated optimization problems in different fields of scientific and engineering disciplines; among them, the particle swarm optimization (PSO) algorithm is one of the most efficient methods. That is why this approach was taken in this research to solve the model in Equation (4.21). The structure of the proposed MOPSO that is based on the PSO algorithm for the multi-objective inventory planning problem at hand is given as follows.

4.4.1 Generating and initializing the particles positions and velocities

PSO was initialized by a group of random particles (solutions) called generation, and then searches for optima by updating generations. The initial population was constructed by randomly generating R particles (similar to the chromosomes of a genetic algorithm). In a d -dimensional search space, let $\vec{x}_k^i = \{x_{k,1}^i, x_{k,2}^i, \dots, x_{k,d}^i\}$ and $\vec{v}_k^i = \{v_{k,1}^i, v_{k,2}^i, \dots, v_{k,d}^i\}$ be, respectively, the position and the velocity of

particle i at time k . Then, Equations (4.22) and (4.23) are applied to generate initial particles, in which x_{\min} and x_{\max} are the lower and the upper bounds on the design variable values and $RAND$ is a random number between 0 and 1.

$$x_0^i = x_{\min} + RAND(x_{\max} - x_{\min}) \quad (4.22)$$

$$v_0^i = x_{\min} + RAND(x_{\max} - x_{\min}) \quad (4.23)$$

An important note for the generating and initializing step of the PSO is that solutions must be feasible and satisfy the constraints. As a result, if a solution vector does not satisfy a constraint, the related vector solution will be penalized by a big penalty on its fitness.

To guarantee the feasibility of the individuals, a penalty function shown in Equation (4.24) was employed to penalize the infeasible solutions those didn't satisfy the constraints.

$$Y(x) = \begin{cases} 0 & \text{if inequality is satisfied} \\ (K(x) - E)^\delta & \text{otherwise} \end{cases} \quad (4.24)$$

In Equation (4.24), δ is the severity of the penalty function (here, $\delta = 10$) and $K(x)$ and E are referred to a typical constraint $K(x) \leq E$.

4.4.2 Selecting the best position and velocity

For every particle, denote the best solution (fitness) that has been achieved so far as:

$$\vec{pbest}_k^i = \{pbest_{k,1}^i, pbest_{k,2}^i, \dots, pbest_{k,d}^i\} \quad (4.25)$$

$$\vec{gbest}_k^i = \{gbest_{k,1}^i, gbest_{k,2}^i, \dots, gbest_{k,d}^i\} \quad (4.26)$$

where \vec{pbest}_k^i in Equation (4.25) is the best position already found by particle i until time k and \vec{gbest}_k^i in Equation (4.26) is the best position already found by a neighbor until time k .

4.4.3 Velocity and position update

The new velocities and positions of the particles for the next fitness evaluation are calculated using Eqs (4.27) and Equation (4.28) (Shayeghi et al., 2009):

$$v_{k+1,d}^i = w \cdot v_{k,d}^i + C_1 \cdot r_1 \cdot (\vec{pbest}_{k,d}^i - x_{k,d}^i) + C_2 \cdot r_2 \cdot (\vec{gbest}_{k,d}^i - x_{k,d}^i) \quad (4.27)$$

$$x_{k+1,d}^i = x_{k,d}^i + v_{k+1,d}^i \quad (4.28)$$

where r_1 and r_2 are random numbers between 0 and 1, coefficients C_1 and C_2 are given acceleration constants towards \vec{pbest} and \vec{gbest} , respectively and w is the inertia weight. Introducing a linearly decreasing inertia weight into the original PSO significantly improves its performance through the parameter study of inertia weight (Naka, Genji, Yura, & Fukuyama, 2001). Moreover, the linear distribution of the inertia weight is expressed as follows (Naka et al., 2001):

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_max} \cdot iteration \quad (4.29)$$

where $iter_max$ is the maximum number of iterations and $iteration$ is the current number of iteration. Equation (4.29) presents how the inertia weight is updated, considering w_{max} and w_{min} are the initial and the final weights, respectively. The parameters $w_{max} = 0.9$ and $w_{min} = 0.4$ that were previously investigated by (Naka et al., 2001) and (Shi & Eberhart, 1999), are used in this research as well.

4.4.4 Stopping criterion

Achieving a predetermined solution, steady-state mean and standard deviations of a solution in several consecutive generations, stopping the algorithm at a certain computer CPU time, or stopping when a maximum number of iterations is reached are usual stopping rules that have been used so far in different research works. In the current research, the MOPSO algorithm stops when the maximum number of iterations was reached.

Figure 4.5 shows the pseudo code of the proposed MOPSO algorithm. Moreover, since the problem and hence the model is new and there is no other available algorithm to compare the results, a multi-objective genetic algorithm (MOGA) is developed in this research for validation and benchmarking. MOGA was coded using roulette wheel in selection operator, population size of 40, uniform crossover with probability of 0.64, one-point random mutation with probability 0.2, and a maximum number of 500 iterations. The computer programs of the MOPSO and MOGA algorithms were developed in MATLAB software and are executed on a computer with 3.80 GHz of core 2 CPU and 4.00 GB of RAM. Furthermore, all the graphical and statistical analyses are performed in MINITAB 15.

```

for i = 1 to Pop
    initialize position(i)
    initialize velocity(i)
    if position(i) and velocity(i) be a feasible candidate solution
        penalty=0
    else penalty= a positive number
    end if
end for
w=[0.4,0.9]
do while Iter <= Gen
    for j=1 to Pop
        Calculate new velocity of the particle
        Calculate new position of the particle
        pbest(iter)=min(pbest(i))
    end for
    gbest(iter)=min(gbest)
    w=Wmax-((Wmax-Wmin)/iter_max)×iter
    modifying the velocity and position of the particle
end while

```

Figure 4.5: The pseudo code of MOPSO algorithm

4.5 The proposed MOGA

The fundamental principle of GAs first was introduced by (Holland, 1992). As mentioned in Chapter 2, many researchers applied and expanded this algorithm in different fields of study. In short, the steps involved in the proposed GA algorithm are:

- i. Setting GA parameters including the crossover probability (P_c), the mutation probability (P_m), population size (PS), and number of generation (NG)
- ii. Initializing the population with the size of PS randomly
- iii. Evaluating the objective function
- iv. Selecting individual for mating pool by roulette wheel selection method and using elitisms
- v. Applying the crossover operation for each pair of chromosomes based on P_c

- vi. Applying mutation operation for each chromosome based on P_m
- vii. Replacing the current population by the resulting new population
- viii. If stopping criteria is met, then stop. Otherwise, go to Step iii.

In order to clarify the steps involved in the GA implementation of this research, only the main features of the algorithm is discussed in more detail.

Basically, for using the most appropriate metaheuristics for solving a problem, the most common way is using the recent ones applied in the same field in the literature. For each problem, the algorithm parameters and also the most suitable operators used in that algorithm are updated and selected in different ways based on the problem need and type. The GA applied to solve the problem proposed in this chapter was tested on the generated problems with different parameters and different operators to get the best solutions where Taguchi method was applied to find the best parameters values of the algorithm to improve the GA performance in the shortest time. Here are also used both Tournament and Roulette Wheel approach individually in selection operator of GA which Roulette Wheel was found with the better efficiency than Tournament approach. Several approaches of crossover operator existed in the literature as well as N-point, Uniform, and so on (Soni & Kumar, 2014; Thakur & Singh, 2014) were also tested on the problem where uniform crossover operator was found to be the best for this problem. Moreover, several types of mutation operator as well as one-point, uniform and two-point were tested on the problem where the one-point mutation approach was chosen as the most efficient approach.

Due to the metaheuristic algorithms are the directed stochastic solution methodologies, they completely depend on the problem types and also the user ability who applies them. Therefore, the users should apply and update the algorithm

based on the proposed problem type to find out the optimal or near optimal solutions. With manipulating the parameters and the operators of these algorithms for each individual problem, the algorithms can improve their performance on the problems as well.

Therefore, the differences between the GA here and the one applied in the previous work are in that the all parameter values and the operators of GA again tested and updated according to the problem type. Moreover, the best parameter values of GA proposed in (Mousavi et al., 2013) were $P_c = 0.8$, $P_m = 0.2$, $Pop = 30$, and $Gen = 700$, while here Taguchi method was applied for each of the generated and also case study problems individually and therefore each problem has its own GA parameters values. Furthermore, in (Mousavi et al., 2013), the elitism operator was used in the GA while here elitism operator was found not to be suitable to improve the GA performance.

The figures and operators of GA used in this study have been also updated accordingly. Figure 4.6 shows the structure of the chromosome generated by the GA.

$$\mathbf{Q} = \begin{pmatrix} Q_{1,1} & Q_{1,2} & \cdots & Q_{1,N-1} \\ Q_{2,1} & Q_{2,2} & \cdots & Q_{2,N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ Q_{m,1} & Q_{m,2} & \cdots & Q_{m,N-1} \end{pmatrix}$$

Figure 4.6: The structure of a chromosome

After testing several kinds of selection approaches existed in the literature as well as Roulette Wheel and Tournament approaches on the problem, the Roulette wheel selection has been chosen as the most appropriate selection process of the

chromosomes in the mating pool. Figure 4.7 depicts the crossover operation of this research to mate pairs of chromosomes to create offspring based on P_c . In Figure 4.7, $Chro_1$ and $Chro_2$ are the parents crossed each time to create Off_1 and Off_2 as offspring. All the infeasible solutions those do not satisfy the constraints will be fined by the penalty function proposed in Eq. (4.24).

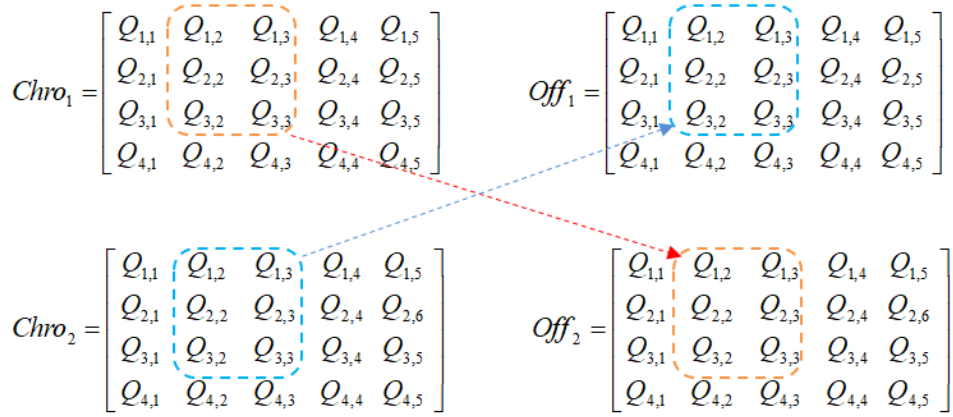


Figure 4.7: The uniform crossover operation

In order to reach better solution of the problem, several different forms of mutation operator provided in the literature are tested. It has been found that the one-point mutation operator was more suitable than the others. In this operations, for each chromosome for which $r_1 > P_c$, select two integer numbers y_1 and y_2 in intervals $[1, m]$ and $[1, N]$, first. Then, generate an integer number randomly between 0 and M_1 for row y_1 and column y_2 of the chromosome. Figure 4.8 shows how to do mutation operator of GA algorithm for a problem with $m = 6$ and $N = 5$ since $y_1 = 3$ and $y_2 = 2$.

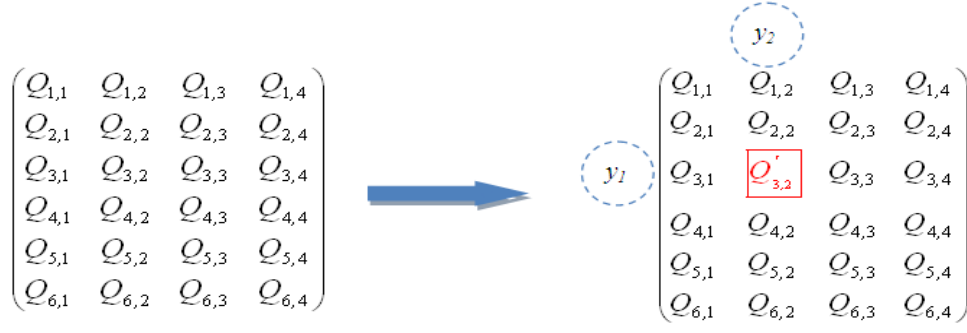


Figure 4.8: The one-point mutation operator

At the end, the proposed GA stops after a predetermined number of iteration obtained by the parameter-tuning procedure discussed in Section 5.

In the next section, some numerical examples are given to illustrate the application of the proposed MOPSO algorithm in real-world environments and to evaluate and compare its performances with the ones obtained by a MOGA method.

4.6 Numerical illustrations

The decision variables in the inventory model Equation (4.21) are $Q_{i,j}$, $X_{i,j}$, $V_{i,j}$ and $b_{i,j}$. We note that the determination of the order quantity of the items in different periods, which are, $Q_{i,j}$, results in the determination of the other decision variables as well. Hence, we first randomly generate $Q_{i,j}$, that is modeled by the particles' position and velocity. Figure 4.9 shows a pictorial representation of the matrix Q for a problem with 4 items in 4 periods, where rows and columns correspond to the items and the periods, respectively.

$$Q_{4,4} = \begin{bmatrix} 124 & 116 & 50 & 0 \\ 205 & 190 & 58 & 0 \\ 114 & 68 & 107 & 0 \\ 43 & 87 & 210 & 0 \end{bmatrix}$$

Figure 4.9: The structure of a particle

4.6.1 The generated data

In order to evaluate the algorithms on the proposed model, due to there is no benchmark in the literature, hence some data are generated where this method is very common in the literature based on the works reviewed in Chapter 2. Table 4.1 depicts the value ranges of the parameters used to generate the different problems to evaluate the algorithms. In other words, in the literature, it is very common for those novel models to generate a wide range of problems while there is no benchmark in the literature to evaluate their proposed algorithms. In addition, in order to clarify how the proposed algorithms perform on different problems with different sizes, the algorithms have been evaluated on a wide range of problems with different sizes to assess their performance in terms of fitness values and CPU time as well. Table 4.2 shows partial data for 40 different problems with different sizes along with their near optimal solutions obtained by MOPSO and MOGA. In these problems, the number of items varies between 1 and 20 and the number of periods takes values between 3 and 15. In addition, the total available budgets and the upper bounds for the order quantities (M_1) are given in Table 4.2 for each problem. From Equation (4.24), in Table 4.2 if a solution doesn't satisfy even one of the constraints (for example here $K(x) \leq E$), the value of $(E - K(x))^\delta$ will be added to the objective functions so that the value of the objective function will become a very big number. Therefore, the both MOGA and MOPSO will remove these infeasible solutions automatically to prevent them to enter the final generation.

In order to illustrate how the results were obtained, consider a typical problem with 5 items and 3 periods (the seventh row in Table 4.2), for which the complete input data is given in Table 4.3. The parameters of the MOPSO and MOGA algorithms are set by Taguchi method where C_1, C_2 the number of population (Pop

) and number of generations (Gen) are the parameters of MOPSO and crossover probability and their level values are shown in Table 4.4. Furthermore, the rest of MOPSO's parameters are set as, $w_{min} = 0.4$, $w_{max} = 0.9$, and the time-periods $T_j = 3$ for $j = 0, 1, 2, 3$. The above parameter settings were obtained by performing intensive runs. Furthermore, the amount of $V_{i,j}$ will be obtained automatically after gaining the order quantity $Q_{i,j}$.

Table 4.1: The input data of the test problems

m	N	M_1	TB	M_2
(1 to 20)	(3 to 20)	(2000 to 115000)	(30000 to 1330000)	(15000 to 2000000)

Table 4.2: Different problems and their optimal TMF values obtained by the two algorithms

Problem No.	m	N	M_1	TB	Objective values		MOPSO					MOGA				
					MOPSO	MOGA	C_1	C_2	Pop	Gen	CPU time	P_C	P_m	Pop	Gen	CPU time
1	1	3	2000	30000	18510	18510	2	2.5	30	200	2.834	0.6	0.1	40	500	4.087
2	2	3	3000	85000	44622	44943	1.5	2.5	20	200	3.132	0.5	0.08	30	200	4.342
3	3	3	5000	170000	85830	86089	1.5	2	30	500	4.237	0.6	0.08	50	300	6.539
4	2	4	3500	130000	92758	93423	2	2	30	100	4.318	0.7	0.2	50	200	6.509
5	3	4	5000	240000	147480	146450	1.5	2.5	40	100	4.904	0.6	0.2	50	200	7.192
6	4	3	6000	260000	132910	133400	2.5	1.5	20	200	5.763	0.6	0.08	40	500	8.122
7	5	3	9000	370000	153840	154550	2	2.5	40	200	6.877	0.6	0.2	50	200	9.213
8	6	3	8800	360000	214620	215510	1.5	2.5	30	500	7.321	0.5	0.1	30	200	11.205
9	7	3	10500	400000	265020	266020	2	2	30	200	8.630	0.7	0.1	40	300	12.093
10	8	5	12000	470000	322850	328250	2.5	2.5	30	500	9.139	0.6	0.08	50	200	13.532
11	8	8	15000	550000	431245	442100	2	2	20	200	11.590	0.7	0.2	30	300	16.442
12	9	5	15000	530000	438790	449835	1.5	2.5	40	100	11.346	0.7	0.2	50	300	17.020
13	9	8	18000	600000	495470	513725	2	2	40	500	13.256	0.7	0.08	50	500	21.324
14	9	9	20000	630000	553276	571250	1.5	2.5	20	100	14.732	0.5	0.1	40	500	22.967
15	10	5	20000	620000	579030	593167	2.5	2.5	20	200	12.874	0.5	0.2	30	200	20.134
16	10	8	25000	700000	642870	665890	1.5	2.5	40	200	15.398	0.7	0.08	30	200	24.321
17	10	10	34000	780000	710035	731280	1.5	1.5	30	200	21.469	0.5	0.08	50	500	32.539
18	10	15	45000	900000	823210	852400	2	2.5	20	500	31.231	0.6	0.1	40	500	46.201
19	11	10	40000	850000	827659	859080	1.5	2.5	30	500	30.137	0.6	0.1	30	500	42.243
20	11	15	45000	870000	867840	897500	2.5	2	40	100	35.092	0.7	0.2	50	200	49.903
21	12	10	48000	870000	902720	948956	1.5	2.5	30	200	34.354	0.5	0.2	50	300	49.320
22	12	12	53000	900000	932760	974380	1.5	2	30	100	36.701	0.6	0.2	40	200	52.430

Table 4.2, Continued

23	12	15	58000	950000	965470	1023950	2	2.5	20	100	42.132	0.5	0.2	30	200	57.810
24	13	10	52000	890000	973200	1043569	1.5	1.5	20	500	40.212	0.7	0.08	40	300	56.192
25	13	13	55000	930000	985439	1089210	1.5	2.5	40	500	42.243	0.6	0.1	30	300	58.111
26	13	15	62000	980000	1056810	1104325	2	2.5	40	500	44.493	0.5	0.1	40	500	62.002
27	15	8	57000	900000	1059835	1110360	2	2	20	200	39.702	0.6	0.08	50	500	58.331
28	15	10	63000	900000	1095430	1176509	2.5	2.5	30	100	42.092	0.6	0.2	30	200	60.217
29	15	12	68000	950000	1198720	1332900	2.5	1.5	30	500	44.620	0.5	0.1	40	200	63.892
30	15	15	75000	1000000	1256980	1447905	1.5	2	20	500	48.410	0.6	0.08	50	200	69.200
31	16	12	70000	940000	1298750	1473400	2	1.5	40	100	45.432	0.7	0.1	50	500	68.793
32	16	15	80000	1050000	1454328	1772349	1.5	2.5	40	500	51.214	0.7	0.2	40	200	74.231
33	17	15	87000	1100000	1543890	1809850	2	2	40	200	55.762	0.7	0.1	50	500	80.321
34	17	17	93000	1140000	1630215	1865780	2.5	2.5	30	500	59.600	0.6	0.1	40	300	87.214
35	18	10	80000	1000000	1678345	1890437	1.5	1.5	30	100	57.320	0.7	0.08	50	300	87.110
36	18	15	98000	1150000	1768950	1924670	2	2.5	20	500	59.739	0.5	0.08	40	300	90.129
37	18	18	103000	1230000	1832450	1987320	1.5	2	30	100	66.002	0.7	0.1	40	200	101.231
38	20	10	100000	1200000	1876895	1998230	2.5	2.5	20	500	60.243	0.6	0.1	50	500	98.772
39	20	15	108000	1260000	1904564	2035689	1.5	2	30	500	69.321	0.7	0.2	50	500	102.197
40	20	20	115000	1330000	1987350	2154670	1.5	1.5	30	100	74.321	0.7	0.08	30	500	109.034
<i>Mean</i>	-	-	-	-	897145	971541	-	-	-	-	31.705	-	-	-	-	46.562
<i>St.Dev</i>	-	-	-	-	581013	652778	-	-	-	-	21.885	-	-	-	-	32.509

Table 4.3: The general data for a problem with 5 items and 3 periods

Product	$D_{i,1}$	$D_{i,2}$	$\pi_{i,1}$	$\pi_{i,2}$	$\hat{\pi}_{i,1}$	$\hat{\pi}_{i,2}$	B_i	H_i	A_i	β_i	S_i
1	1200	800	20	18	9	10	3	5	20	0.5	4
2	1300	900	20	18	9	10	7	5	15	0.5	6
3	1500	1200	11	14	8	12	5	6	25	0.8	7
4	2100	2000	11	14	8	12	8	6	18	0.8	5
5	1800	1600	12	15	9	11	7	7	19	0.6	6

Table 4.4: The parameters of the two algorithms and their levels

Algorithms	Factors	Levels $[I,2,3]$
MOPSO	$C_1(A)$	[1.5,2,2.5]
	$C_2(B)$	[1.5,2,2.5]
	$Pop(C)$	[20,30,40]
	$Gen(D)$	[100,200,500]
MOGA	$P_C(A)$	[0.5,0.6,0.7]
	$P_m(B)$	[0.08,0.1,0.2]
	$Pop(C)$	[30,40,50]
	$Gen(D)$	[200,300,500]

The prices of each item in each period under discount policies proposed in Table 4.3 are shown as follows:

$$P_1 = \begin{cases} 21 & 0 < Q \leq 950 \\ 20 & Q > 950 \end{cases}$$

$$P_2 = \begin{cases} 21 & 0 < Q \leq 60 \\ 19 & 60 < Q \leq 110 \\ 18 & Q > 110 \end{cases}$$

$$P_3 = \begin{cases} 25 & 0 < Q \leq 90 \\ 23 & Q > 90 \end{cases}$$

$$P_4 = \begin{cases} 17 & 0 < Q \leq 120 \\ 15 & Q > 120 \end{cases}$$

$$P_5 = \begin{cases} 16 & 0 < Q \leq 100 \\ 12 & Q > 100 \end{cases}$$

In Table 4.3, the general data for Problem No. 7 of Table 4.2 with 5 items and 3 time periods (seasons) is shown. The demand for all five items in the first period is shown by $D_{i,1}$ and the demand for all five items in the second period is indicated by $D_{i,2}$ ($i = 1, 2, 3, 4, 5$). Due to the system is closed in the last period no demand for any item is accepted (i.e. $D_{i,3}=0$). To be clear, in Table 4.3 for example the demand for item 1 in the first period (season) is equal to 1200 with price 20 while the demand for this item in the second period will be 800 with price 21 where the price of this item is proposed under AUD policy as follows:

$$P_1 = \begin{cases} 21 & 0 < Q \leq 950 \\ 20 & Q > 950 \end{cases}$$

Tables 4.7 and 4.8 show the optimal solutions as well as the optimal order quantities for satisfying the demands for each item in each period of Table 4.3 obtained by MOPSO and MOGA, respectively. To be clarified, in Table 4.7, the optimal order quantity of item 1 in the first period (season) is equal to 1215 items ($Q_{1,1}=1215$) to satisfy 1200 demands received in this period ($D_{1,1}=1200$) while 15 out of these items are kept in the storage (

$X_{1,2}=15$) to satisfy the demands will receive in the second period. Therefore, there is no demand for item 1 left unsatisfied in the first period. In other words, there is no shortage quantity of item 1 in the second period ($b_{1,2}=0$) while all the demands for item 1 are satisfied in this period. From Table 4.7, the optimal order quantity of item 1 for period 2 is equal to 159 ($Q_{1,2}=159$) to satisfy 800 demands received in the second period ($D_{1,2}=800$). The order quantity of item 1 in addition to the inventory left from the first period mines the demand will get the shortage amount for next period ($Q_{1,2} + X_{1,2} - D_{1,2} = -626$), then we have $X_{1,3}=0$ and $b_{1,3} = 626$. It will be explainable in the same way for other items.

The weights associated with the objectives are as triangular fuzzy number $w = [w_a, w_b, w_c]$ shown in Figure 4.10 where membership function of variable x is given by Equation (4.30).

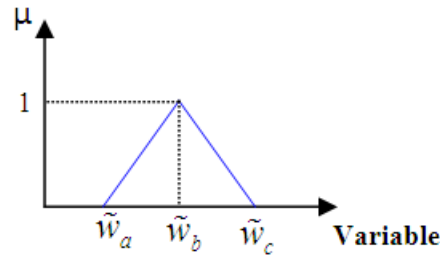


Figure 4.10: The triangular fuzzy numbers

$$\mu(x) = \begin{cases} 0 & x < w_a \\ \frac{x - w_a}{w_b - w_a} & w_a < x < w_b \\ \frac{w_c - x}{w_c - w_b} & w_b < x < w_c \\ 0 & w_c < x \end{cases} \quad (4.30)$$

Now, in order to get crisp interval by α -cut operation, interval \tilde{w}_a can be obtained as follows ($\forall \alpha \in [0,1]$)

$$\frac{w_a^{(\alpha)} - w_a}{w_b - w_a} = \alpha, \frac{w_c - w_c^{(\alpha)}}{w_c - w_b} = \alpha \quad (4.31)$$

We have $w_a^{(\alpha)} = (w_b - w_a)\alpha + w_a$; $w_c^{(\alpha)} = w_c - (w_c - w_b)\alpha$

Therefore, $w_\alpha = [w_a^{(\alpha)}, w_c^{(\alpha)}] = [(w_b - w_a)\alpha + w_a, w_c - (w_c - w_b)\alpha]$. While the weights of the objectives of the proposed problem are fuzzy numbers the best approach to convert the fuzzy numbers to crisp ones will be α -cut approach according to the literature reviewed in Chapter 2. In other words, if α -cut approach is not used the results of the model will be remained as fuzzy numbers which will be useless.

where, in all the generated data $w_1 = 1 - w_2$, $w_2 = [0.2, 0.3, 0.6]$ and $\alpha = 0.5$.

In other words, while the weights allocated to each objective are uncertain in the real world, in this study these weights are considered as triangular fuzzy numbers. One of the common methods to convert (decompose) the fuzzy numbers to crisp (deterministic) numbers is α -cut operation which is expressed by formula proposed in Equation (4.31).

In this study, using the α -cut method shown in Equation (4.31) and triangular fuzzy numbers $w_2 = [0.2, 0.3, 0.6]$ and $w_1 = 1 - w_2$ and $\alpha = 0.5$, we have: $w_{0.5} = [0.25, 0.45]$ which is shown by Figure 4.11.

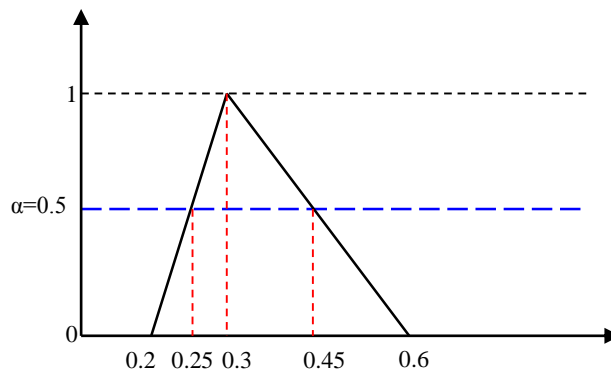


Figure 4.11: The α -cut ($\alpha = 0.5$) procedure applied for the proposed problems with triangular fuzzy numbers

In this research, α -cut and centroid defuzzification method are used by MATLAB 2013a to make crisp numbers. Therefore, after using these methods we have $w_2 = 0.3667$ and $w_1 = 1 - w_2 = 0.6333$.

To perform Taguchi approach in this paper, a L_9 design is utilized, based on which the results for Problem No. 7 described in Table 4.2 are shown in Table 4.5 as an example. The optimal values of the levels of the algorithms' parameters shown in Table 4.5 were presented in Table 4.6. Figure 4.12 depicts the mean S/N ratio plot each level of the factors of MOPSO and MOGA for Problem 7 in Table 4.2.

Table 4.5: The Taguchi L_9 design along with objective values of the algorithms

Run no.	A	B	C	D	MOPSO	MOGA
1	1	1	1	1	154040	154980
2	1	2	2	2	154367	154760
3	1	3	3	3	154220	155075
4	2	1	2	3	153944	154875
5	2	2	3	1	153985	155230
6	2	3	1	2	154568	155102
7	3	1	3	2	154215	154780
8	3	2	1	3	154320	154750
9	3	3	2	1	154100	155111

Table 4.6: The optimal levels of the algorithms' parameters for Problem 7 of Table 4.2

Algorithms	Factors	Optimal levels
MOPSO	C_1	2
	C_2	2.5
	Pop	40
	Gen	200
MOGA	P_c	0.6
	P_m	0.2
	Pop	50
	Gen	200

Tables 4.7 and 4.8 show the best result obtained by MOPSO and MOGA for the problem with 5 items and 3 periods (Problem 7), respectively including the amounts of decision variables and the optimal objective values. In these Tables, TMF is the best value of the bi-objective inventory problem, which is given in the last two columns of Tables 4.7 and 4.8. Similarly, the best values of TMF for the other problems were obtained and are summarized in the sixth and seventh columns of Table 4.2.

To compare the performances of the MOPSO and MOGA, several statistical and graphical approaches are employed. A one-way ANOVA analysis of the means of the algorithms in confidence 0.95% is used to compare and evaluate the objective values of the generated 40 problems. Table 4.9 shows the ANOVA analysis of the results of the two algorithms that demonstrates no difference between both algorithms. Moreover, the mean and standard deviation (*Std.Dev*) of the objective values of the 30 generated problems shows that the MOPSO has the better performance in terms of the objective values in comparison with the MOGA. In addition, a pictorial presentation of the performances of the two algorithms shown by Figure 4.13 displays that the MOPSO is more efficient than the MOGA algorithm in the large number of the problems.

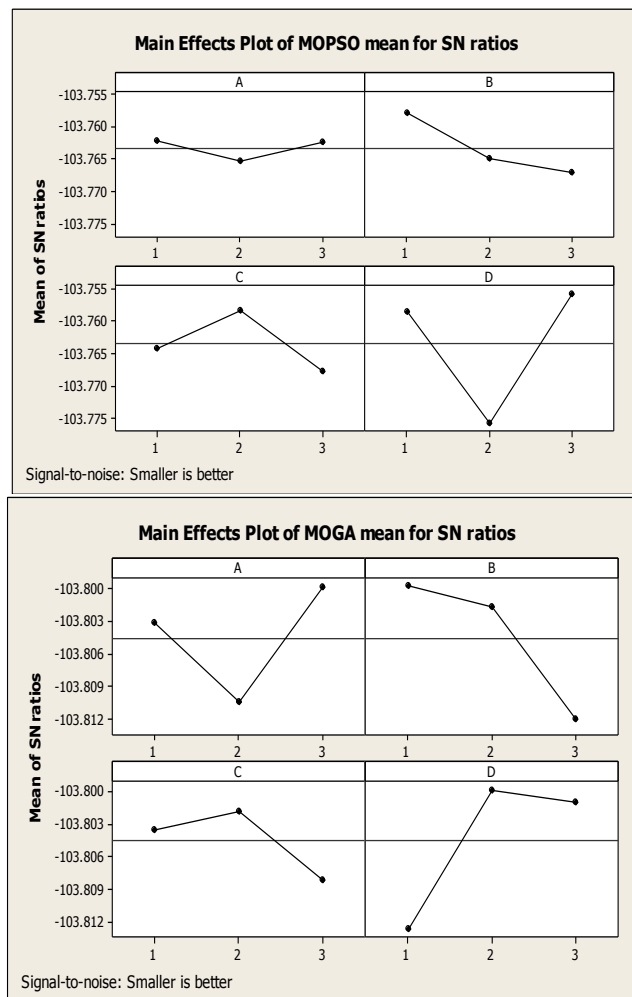


Figure 4.12: The mean S/N ratio plot for parameter levels of MOPSO and MOGA in Problem 7 of Table 4.2

Table 4.7: The best result of the MOPSO algorithm

Product	$Q_{i,1}$	$Q_{i,2}$	$X_{i,2}$	$X_{i,3}$	$V_{i,1}$	$V_{i,2}$	$b_{i,1}$	$b_{i,2}$	TMF
1	1215	159	15	0	405	53	0	626	133958
2	1162	252	0	0	166	36	138	648	
3	1555	190	55	0	311	38	0	955	
4	1360	864	0	0	170	108	740	1136	
5	1435	420	0	0	205	60	365	1180	

Table 4.8: The best result of the MOGA algorithm

Product	$Q_{i,1}$	$Q_{i,2}$	$X_{i,2}$	$X_{i,3}$	$V_{i,1}$	$V_{i,2}$	$b_{i,1}$	$b_{i,2}$	TMF
1	1221	168	21	0	407	56	0	611	151525
2	959	392	0	0	137	56	341	508	
3	1220	390	0	0	244	78	280	810	
4	1168	960	0	0	146	120	932	1040	
5	168	2254	0	654	24	322	1632	0	

Table 4.9: The ANOVA analysis of the optimal inventory costs for the algorithms

Source	DF	SS	MS	F	P-value
Factor	1	1.11E+11	1.11E+11	0.28	0.6
Error	78	3.11E+13	3.99E+11	-	-
Total	79	3.12E+13	-	-	-

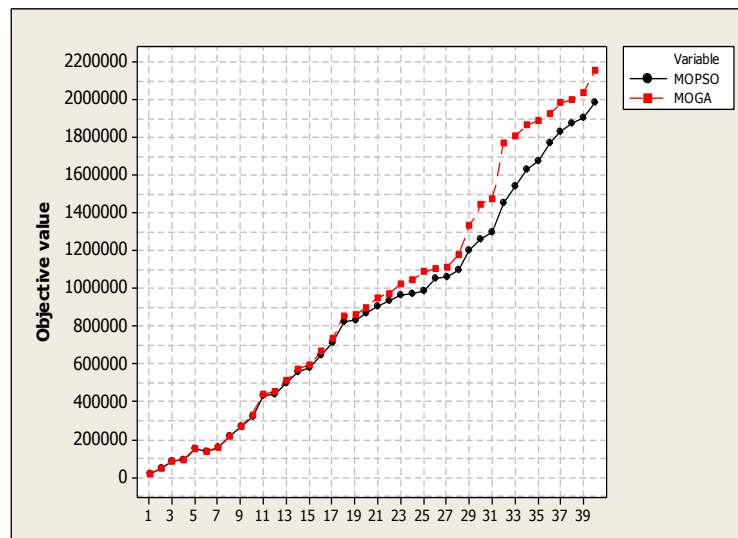
**Figure 4.13:** The pictorial representation of the performances of the algorithms

Figure 4.14 depicts the Box-plot and the individual value plot and Figure 4.15 shows the residual plots for the algorithms.

A comparison of the results in Table 4.2 shows that the MOPSO algorithm performs better than the MOGA in terms of the fitness functions and also CPU time values.

Figure 4.16 shows the t-test (ANOVA) for the means of both MOPSO and MOGA performance in terms of CPU time where the results are in favor of MOPSO. In other words, MOPSO is faster than MOGA in finding the optimal solutions in the proposed 40 different problems shown in Table 4.2.

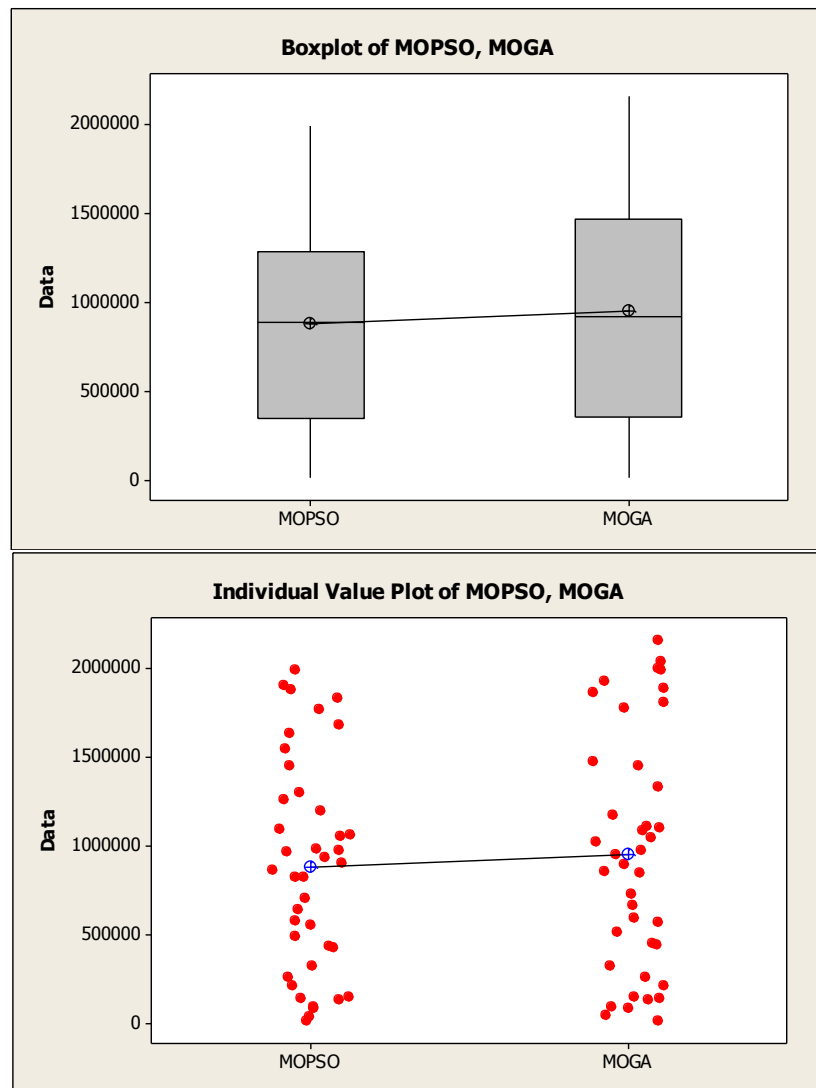


Figure 4.14: The Box-plot and the individual value plot of the performances of the algorithms

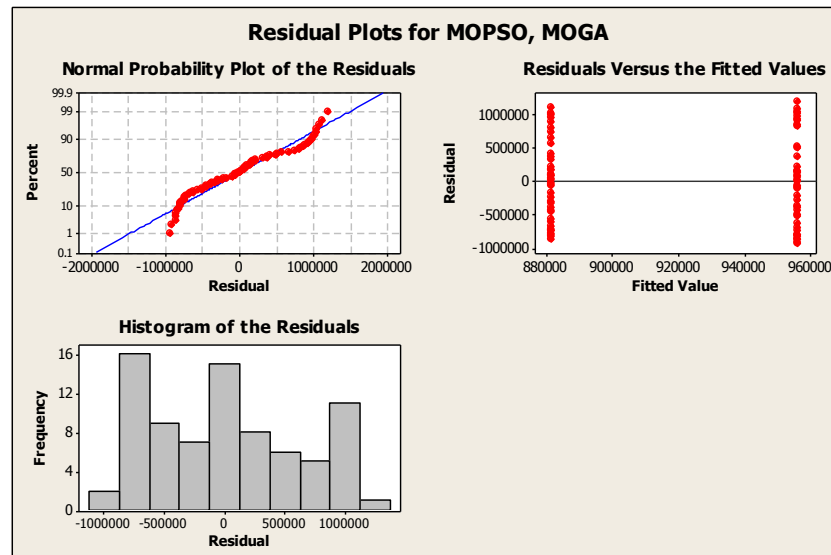


Figure 4.15: The residual plots of the algorithms

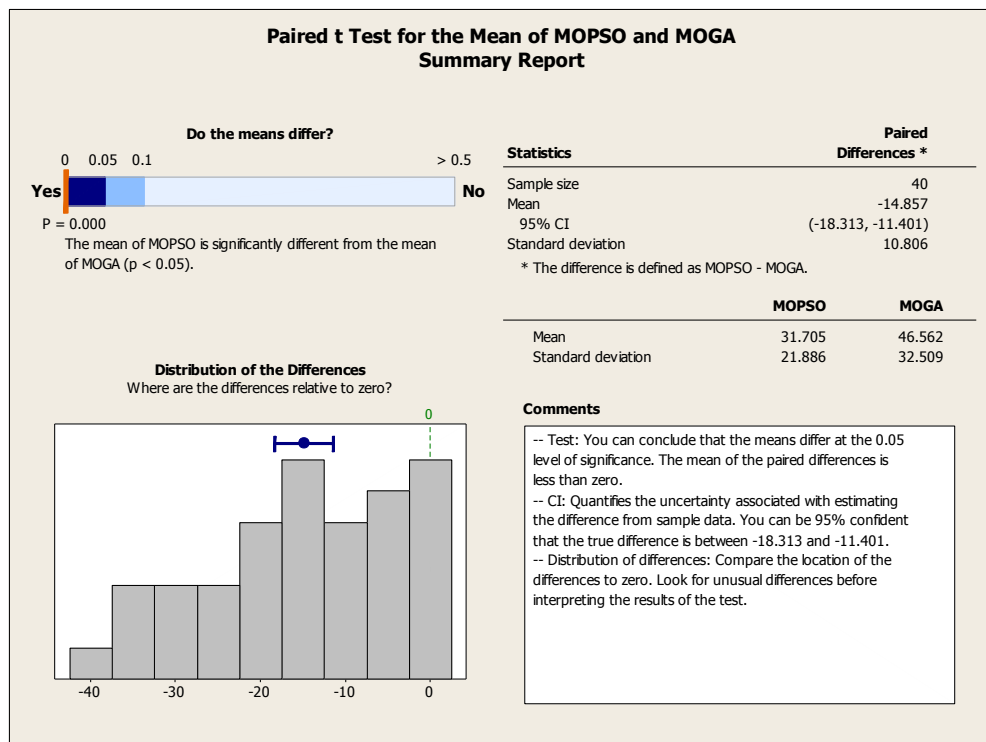


Figure 4.16: The ANOVA analysis of the CPU time for both the algorithms

In Table 4.7, the details of the costs and TMF obtained by MOPSO are listed as:

Total Ordering cost = 291 \$

Total Holding cost = 25,775 \$

Total Shortage cost = 8444 \$ includes 6408 \$ backordering cost and 2036 \$ lost sale

Total Purchasing cost = 148687 \$

Z_1 = Total costs = 183196 \$, Z_2 = 48890 m², TMF = 133958

In Table 4.8, the details of the costs and TMF obtained by MOGA are listed as:

Total Ordering cost = 291 \$

Total Holding cost = 35,954 \$

Total Shortage cost = 21564 \$ includes 15666 \$ backordering cost and 5898 \$ lost sale

Total Purchasing cost = 150112 \$

Z_1 = Total costs = 207911 \$, Z_2 = 54112 m², TMF = 151525

4.6.2 A Case study

In order to test and evaluate the model on a real case, some data were collected from a wood industry manufacturing-trade company called ORAMAN WOOD Industry (Alikar Wood Industry (AWI)) located in Iran manufacturing High Gloss products for building some finished items as well as cabinet, decoration board, shelf and etc. The company was established in 2003 in a plot with 10000 m² with only one production line (Finished foils) which today has been extended to several other wood products as well as High Gloss. The company had more than 100 personnel including 32 experts and about 70 workers. Today, the company supports almost the most part of Iran in addition to some of Middle-East countries.

To produce High Gloss sheets, the company imports several raw material i.e. MDF sheets, Hot Melt Adhesive (HMA), High Gloss film (HGF) and PVC from overseas. All the items were imported from Malaysia, Thailand, Turkey, China and Germany. The company plans to order the materials every three months (the replenishment cycle is three months). Furthermore, the planning horizon is going to be one year or 12 months. The

data collected from the company is shown in Table 4.10 where a problem with four items and three periods is considered. To validate the data, the real data sheet collected from the company is shown in Appendix A. The demand values of the customers are almost deterministic and known and have different values in some periods than each other.

The company faces shortages in some periods because of transportation and sanction issues. The values of the other parameters were set as follows (*for $i = MDF, adhesive, HGF, PVC$) and $(j = 1, 2, 3)$, $M_1 = 200$, $M_2 = 700$ and $TB = 120,000,000 IT$. The MDFs are delivered into the boxes each includes 20 MDF ($B_{MDF} = 20$), adhesives are delivered into the boxes each includes 18 Kg ($B_{HMA} = 18$), HGF are delivered into the boxes each includes 4 pieces ($B_{HGF} = 4$) and PVC delivered into the boxes each includes 6 pieces ($B_{PVC} = 8$).*

The owner of the company wants to obtain minimum of the total costs endured on the company by the matters of inventory system. Furthermore, due to assigning the current storage to store the items produced by other production lines, for storing the imported products as well as *MDF, adhesive, HGF* and *PVC* , the company needs to build a new storage where the owner is looking for the optimal storage space (m^2) required for this matter.

Table 4.10: The collected data for the case study with 4 items and 4 periods

Product	$D_{i,1}$	$D_{i,2}$	$D_{i,3}$	$\pi_{i,2}$	$\pi_{i,3}$	$\pi_{i,4}$	$\hat{\pi}_{i,2}$	$\hat{\pi}_{i,3}$	$\hat{\pi}_{i,4}$	H_i	A_i	S_i
<i>MDF (sheet)</i>	110	110	180	50000	60000	70000	80000	90000	60000	1000	3000000	500
<i>HMA (Kg)</i>	50	50	60	50000	60000	80000	70000	90000	50000	100	150000	100
<i>HGF (120m)</i>	32	32	50	30000	40000	60000	80000	100000	90000	500	1400000	200
<i>PVC (120m)</i>	32	32	50	40000	20000	40000	160000	40000	60000	500	240000	100

The data shown by Table 4.10 was collected from ORAMAN Company in 2016 where the orders were made each three months started from January 2016 to January 2017.

The company purchased all the items (MDF, HMA, HGF and PVC) under AUD policy provided by the suppliers with the following prices:

$$P_{MDF} = \begin{cases} 53000 & 0 < Q \leq 100 \\ 52000 & Q > 100 \end{cases}$$

$$P_{HMA} = 30000$$

$$P_{HGF} = \begin{cases} 810000 & 0 < Q \leq 70 \\ 800000 & Q > 70 \end{cases}$$

$$P_{PVC} = \begin{cases} 160000 & 0 < Q \leq 70 \\ 150000 & Q > 70 \end{cases}$$

After parameter tuning using Taguchi method, the optimal parameters of MOPSO are $C_1=1.5$, $C_2=2.5$, $Pop=50$ and $Gen=500$ while the optimal parameters of MOGA are $P_c=0.5$, $P_m=0.1$, $Pop=30$ and $Gen=500$.

The optimal solutions obtained by both MOGA and MOPSO are shown by Tables 4.11 and 4.12, respectively. The total ordering costs according to Iran currency (Iran Toman (IT)) obtained by both algorithms are identical and equal to

$$\text{Total Ordering Cost (TOC)} = 14,370,000 \text{ IT}$$

The other costs obtained by MOGA are listed as follows:

$$\text{Total Holding Cost (THC)} = 994,400 \text{ IT}$$

$$\text{Total Shortage Cost (TSC)} = 0$$

$$\text{Total Purchasing Cost (TPC)} = 353,720,000 \text{ IT}$$

Therefore, the total costs spent by the company which are obtained by MOGA are calculated as

$$Z_1 = \text{TOC} + \text{THC} + \text{TSC} + \text{TPC} = 396,084,400 \text{ IT.}$$

The results obtained by MOGA show that there is no shortage for all items in each period (i.e. $b_{i,j} = 0$ (for $i = \text{MDF}, \text{HMA}, \text{HGF}$; for $j=1, 2, 3, 4$)). The duration of each period is equal to 1 (i.e. $T_j - T_{j-1} = 1$ for $j=1, 2, 3, 4$).

To calculate the required storage space, each 200 sheets of MDF, each 250 Kg of adhesive, each 100 rolls of High Gloss film and also each 100 rolls of PVC are accumulated in the storage height. Therefore, the total required space to store the ordered items calculated by MOGA is as $Z_2 = 4,550 \text{ m}^2$.

Since the preference of the manager of the company for both objectives Z_1 and Z_2 are identical (i.e. $W_1 = W_2 = 0.5$), hence $\text{TMF} = 65,744,000$. The time elapsed by MOGA to find the optimal solutions is 8.624 Seconds.

Furthermore, the total costs obtained by MOPSO under the optimal solutions shown by Table 4.12 are listed as follows:

The Total Ordering Cost (TOC) is equal to 14,370,000 IT, the Total Holding Cost (THC) is equal to 1,169,900 IT. In case of shortages, 10 percent of the customers of the company facing the shortage will leave the system without receiving their orders (i.e. $\beta_i = 0.9$). Total Shortage Cost (TSC) = 1,097,000 where the total backorder costs are equal to 972,000 IT and the total lost sale costs are equal to 125,000 IT. The Total Purchasing Cost (TPC) is equal to 114,020,000 IT.

Therefore, the total costs spent by the company which are obtained by MOPSO are calculated as

$$Z_1 = \text{TOC} + \text{THC} + \text{TSC} + \text{TPC} = 130,656,900 \text{ IT.}$$

Moreover, the total required space for storing the order quantities obtained by MOPSO is calculated as $Z_2 = 3,060 \text{ m}^2$.

Since the preference of the manager of the company on the both objectives Z_1 and Z_2 are identical (i.e. $W_1 = W_2 = 0.5$), hence the total costs obtained by MOPSO

is $TMF = 65,330,000$. The time elapsed by MOPSO to find the optimal solutions is 6.711 Seconds.

Therefore, the results obtained by both MOGA and MOPSO on the case study show that MOPSO is a powerful and suitable algorithm to solve the problem in comparison with MOGA in terms of TMF and CPU time.

4.7 Summary

In this chapter, a bi-objective multi-item multi-period inventory problem with total available budget under all unit discount for some items and incremental quantity discount for other items was considered. The orders were assumed to be placed in batch sizes and the order quantities at the end period were zeroes. Shortages were allowed and contained backorder and lost sale. It was assumed that the beginning order quantity in primary period was zeroes and the order quantity in each period was more than the shortage quantity in the previous period. Due to adopting decisions related to a certain department of production planning (extending warehouse or building a new manufacturing line), the manager decided to build a new warehouse for the ordering items. The objectives were to minimize both the total inventory costs and the total required storage space, for which a weighted combination was defined the objective function. The aim of the study was to determine the optimal order quantity and the shortage quantity of each product in each period such that the objective function is minimized and the constraints hold. The developed model of the problem was shown to be an integer nonlinear programming mixed with binary variables. To solve the model, both a multi-objective particle swarm optimization and multi-objective genetic algorithms were applied. The results showed that for the 10 specific problems the MOPSO performs better than, the MOGA in terms of the fitness function values.

Table 4.11: The optimal solutions obtained by MOGA for the case study

Product	$Q_{i,1}$	$Q_{i,2}$	$Q_{i,3}$	$Q_{i,4}$	$X_{i,2}$	$X_{i,3}$	$X_{i,4}$	$V_{i,1}$	$V_{i,2}$	$V_{i,3}$	$V_{i,4}$	$b_{i,1}$	$b_{i,2}$	$b_{i,3}$	$b_{i,4}$
<i>MDF</i>	160	160	240	0	50	100	160	8	8	12	0	0	0	0	0
<i>HMA</i>	72	90	108	0	22	62	110	4	5	6	0	0	0	0	0
<i>HGF</i>	80	100	150	0	48	116	216	20	25	30	0	0	0	0	0
<i>PVC</i>	80	120	150	0	48	136	236	10	15	19	0	0	0	0	0

Table 4.12: The optimal solutions obtained by MOPSO for the case study

Product	$Q_{i,1}$	$Q_{i,2}$	$Q_{i,3}$	$Q_{i,4}$	$X_{i,2}$	$X_{i,3}$	$X_{i,4}$	$V_{i,1}$	$V_{i,2}$	$V_{i,3}$	$V_{i,4}$	$b_{i,1}$	$b_{i,2}$	$b_{i,3}$	$b_{i,4}$
<i>MDF</i>	140	100	140	0	30	20	0	7	5	7	0	0	0	0	20
<i>HMA</i>	54	36	180	0	4	0	2	3	2	10	0	0	0	10	0
<i>HGF</i>	40	20	192	0	8	0	6	10	5	48	0	0	0	8	0
<i>PVC</i>	88	176	176	0	0	8	46	11	22	22	0	0	0	0	0

CHAPTER 5: A MODIFIED PARTICLE SWARM OPTIMIZATION FOR SOLVING THE INTEGRATED LOCATION AND INVENTORY CONTROL PROBLEMS IN A TWO-ECHELON SUPPLY CHAIN NETWORK

5.1 Introduction

This Chapter considers a supply chain network in which several vendors (manufacturers) are considered to be located in a certain area among numerous buyers who own the warehouses and have limited capacity. Furthermore, the objective is to find the optimal quantity that each buyer orders from the vendors. In a location allocation problem, several new facilities are located in between a number of pre-specific customers in a determined area such that the total transportation cost from facilities to customers is minimized. In order to optimize the proposed multi-product multi-period supply chain problem and determine the location of the vendors in a specific area among the buyers, two meta-heuristic algorithms which are a modified particle swarm optimization (MPSO) and GA were utilized in this Chapter.

The rest of the chapter is organized as follows. In section 5.2, a description of the provided problem is explained where the supply chain model was formulated. Section 5.3 contains the solution methodologies of the problem in which MPSO and GA were explained. In section 5.4, some numerical examples were generated to compare the algorithms where Taguchi approach is presented for setting the algorithms' parameters. Finally, a conclusion of the problem was represented in section 5.5.

5.2 Problem description and formulation

This chapter aims to optimize a supply chain network for a multi-product multi-period inventory system with multiple buyers, multiple vendors and warehouses with limited capacity owned by the vendors. The inventory replenishment starts at a certain time-period and finish at another time-period where the buyers purchase the products from the vendors during these interval periods. The vendors provide (produce) the various products to the buyers with variable demand rates under all-unit discount policy since the production capacity of each vendor is restricted. The vendors satisfy the buyers' demands immediately in all the periods so that no shortages occur during the replenishment. When the demands of the buyers are satisfied in a period, the products remained from the period in addition to the order quantities of the next period enter into the warehouses.

The total available budget for purchasing the products and also the total vendors' warehouse space were constrained. Moreover, the distance between the buyers and the vendors is assumed as Euclidean distance. Figure 5.1 shows a graphical illustration of the proposed model where the vendors store the products into their warehouses and then the products are transported from these warehouses to the buyers.

The model proposed in this chapter is an extension of the inventory model shown in Chapter 4 in which a novel mathematical model of the supply chain network of the proposed multi-item multi-period (seasonal) inventory problem is considered. In the problem provided in Chapter 4, the inventory system of multiple items in a finite horizon (multi-period) was formulated without considering where these items come from and go to. In other words, the members (upstream and downstream) of the system were not considered. In this chapter, a supply chain network of multiple item produced and distributed in each period (season) in a finite horizon is considered

where the manufacturers produce the different items and then store them into their own storages to meet the demand of the customers.

Therefore, the contributions of the work presented here are as follows. This is for the first time in the literature a novel model of supply chain network of multi-item multi-period (seasonal) inventory control problem is formulated at which manufacturers and buyers are involved. The production process of manufacturers is also considered in this work where the production capacity of the items is restricted. The owner of the supply chain network aims to find the coordinates of the optimal locations of the manufacturers with subject to the customers' demands and the order quantities received from these customers so that the total system costs are minimized. The items are delivered from the manufacturers' storages with limited space to customers using trucks with limited capacity where the path (distance) between each manufacturer and customer follows Euclidean distance function. In order to encourage the customers to stay with the company, the manufacturers propose their items under AUD discount policy. Due to the complexity of the problem (being Np-hard), while the normal solvers are unable to solve the problem, a modified version of the PSO (MPSO) with different type of representation of solutions is used. Furthermore, a case study is also performed in a real company to evaluate the model on a real problem.

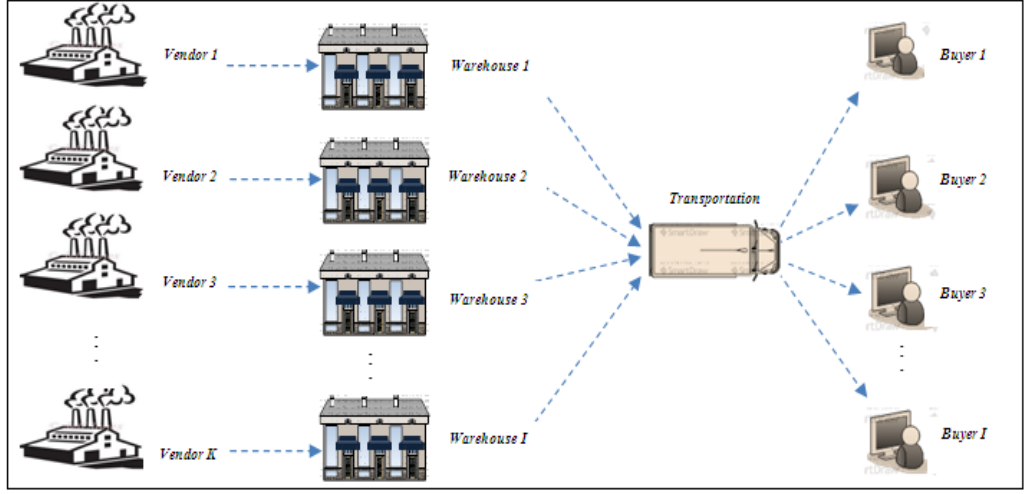


Figure 5.1: The illustration of the supply chain model

5.2.1 Notations and assumptions

The indexes, notations and assumptions involved in the supply chain model come as follows:

(a) *Indexes*

$i = 1, 2, \dots, I$ is the index of the buyers

$j = 1, 2, \dots, J$ is the index of the products

$k = 1, 2, \dots, K$ is the index of the vendors

$t, t = 0, 1, \dots, T$ is the index of the time periods

$p, p = 1, 2, \dots, P$ is the index of the price break points

(b) *Notations*

u_{ijktp} : A binary variable that is set to 1 if buyer i purchases product j from vendor k at price break point p in period t , and set to 0 otherwise

d_{ijkt} : Demand of buyer i for product j produced by vendor k in period t

T_{ijkt} : Total time elapsed up to and including the t^{th} replenishment cycle of the j^{th} product ordered by buyer i from vendor k

$f_k :$	The production capacity of vendor k
$S_k :$	The storage capacity of vendor k
$h_{ijkt} :$	Inventory holding cost per unit of j^{th} product in the warehouse owned by vendor k sold to buyer i in period t
$A_{ijkt} :$	Ordering cost (transportation cost) per unit of j^{th} product from vendor k to buyer i in period t
$c_{ijkt} :$	Purchasing cost per unit of j^{th} product paid by buyer i to vendor k at p^{th} price break point in period t
$e_{ijkt} :$	p^{th} price break-point proposed by vendor k to buyer i for purchasing j^{th} product in period t ($e_{ijktl} = 0$)
$s_{ijkt} :$	The required warehouse space for vendor k to store per unit of j^{th} product sold to buyer i in period t
$B :$	The total available budget
$C :$	An upper bound for the available order quantity
$w_{ijkt} :$	A binary variable that is set to 1 if buyer i orders product j from vendor k in period t , and set to 0 otherwise
$TC :$	The total inventory costs
$a_i = (a_{i1}, a_{i2}) :$	The coordinates of the location of buyer i
$y_k = (y_{1k}, y_{2k}) :$	The coordinates of the location of vendor k (decision variable)
$Q_{ijkt} :$	Ordering quantity of j^{th} product purchased by buyer i from vendor k in period t (decision variable)

- x_{ijkt} : The initial (remained) positive inventory of j^{th} product purchased by buyer i from vendor k in period t ($x_{ijk1} = 0$) (decision variable)
- $I(t)$: Inventory position in period t
- $g(y_{1k}, y_{2k})$: The region to locate the vendor k (here is trapezoidal)

(c) Assumptions

- i. The shortages are not allowed
- ii. The Replenishments are instantaneous
- iii. The buyers' demand rates of all products are independent from each other and variable in the different periods.
- iv. The initial order quantity of all products ordered by the buyers from each vendor is zero (i.e., $x_{ijk1} = 0$)
- v. The order quantity of the products from the vendors made by the buyers in each period is at least equal to the demand rates in during the period (i.e. $Q_{ijkt} \geq d_{ijkt} T_{ijkt}$).
- vi. Planning horizon is finite and known. In the planning horizon, there are T periods.
- vii. The total available budget to purchase the products, the total warehouse space of each vendor and the total production capacity of the vendors are limited.
- viii. The distance between the buyers and the vendors is assumed to be Euclidean distance.
- ix. The vendors have the limited capacity of producing the items.

- x. The paths between the buyers and the vendors are connected and the unit transportation cost is proportionate of the quantity supplied and the travel distance.
- xi. Each vendor has their own warehouse to keep the produced items before reaching the demands
- xii. No order is made at the last period

5.2.2 Problem formulation

In order to formulate the supply chain problem in hand, Figure 5.2 provides some possible scenarios for the inventory system of the problem. The objective function of the problem is to minimize the total costs comprising Transportation cost (TrC), Holding cost (HC) and Purchasing cost (PC). To formulate the objective function, first the transportation cost is calculated. The transportation cost is obtained by considering the Euclidean distance between the buyers and the vendors using the following equation:

$$TrC = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T Q_{ijkt} w_{ijkt} A_{ijkt} \sqrt{(y_{k1} - a_{i1})^2 + (y_{k2} - a_{i2})^2} \quad (5.1)$$

where $\sqrt{(y_{1k} - a_{i1})^2 + (y_{2k} - a_{i2})^2}$ calculates the Euclidean distance between buyer i with coordinate $a_i = (a_{i1}, a_{i2})$ and vendor k with coordinate $y_k = (y_{1k}, y_{2k})$.

According to Figure 4.2, the holding cost in interval $[T, T-1]$ is obtained using the following equation:

$$\int_{T-1}^T I(t) dt \quad (5.2)$$

and for the whole periods we have:

$$\sum_{t=2}^T \int_{t-1}^t I(t) dt \quad (5.3)$$

Therefore, the total holding cost becomes:

$$HC = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} (x_{ijkt} + Q_{ijkt} + x_{ijkt+1}) T_{ijkt} h_{ijkt} / 2 \quad (5.4)$$

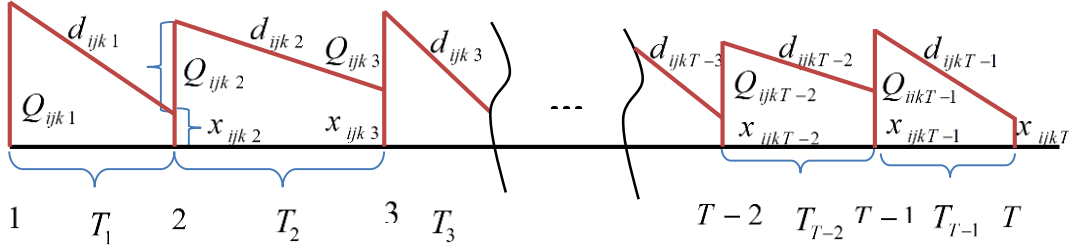


Figure 5.2: Some possible scenarios for the available inventory system

In the proposed supply chain problem, the buyers purchase the products in each period under the discount strategy provided by the vendors. In this work, the products are bought under AUD policy since the price-break point suggested by the vendors is as:

$$\begin{cases} c_{ijkt1} & e_{ijkt1} \leq Q_{ijkt} < e_{ijkt2} \\ c_{ijkt2} & e_{ijkt2} \leq Q_{ijkt} < e_{ijkt3} \\ & \vdots \\ c_{ijktP} & e_{ijktP} \leq Q_{ijkt} \end{cases} \quad (5.5)$$

Then, the total purchasing cost under AUD policy is obtained as Equation (5.6).

$$PC = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \sum_{p=1}^P Q_{ijkt} c_{ijktP} u_{ijktP} \quad (5.6)$$

Therefore, the objective function of the total cost comes as:

$$Tc = TrC + HC + PC \quad (5.7)$$

The supply chain model proposed in this paper is formulated as follows:

$$\begin{aligned}
MinTc = & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T Q_{ijkt} A_{ijkt} w_{ijkt} \sqrt{(y_{1k} - a_{i1})^2 + (y_{2k} - a_{i2})^2} + \\
& (\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} (x_{ijkt} + Q_{ijkt} + x_{ijk,t+1}) T_{ijkt} h_{ijkt} / 2) + \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \sum_{p=1}^P Q_{ijkt} c_{ijktp} u_{ijktp}
\end{aligned} \tag{5.8}$$

S.t.

$$x_{ijk,t+1} = x_{ijkt} + Q_{ijkt} - d_{ijkt} T_{ijkt} \tag{5.8-1}$$

$$\sum_{i=1}^I \sum_{j=1}^J (Q_{ijkt} + x_{ijkt}) s_{ijkt} \leq S_k \tag{5.8-2}$$

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{t=1}^T Q_{ijkt} \leq f_k \tag{5.8-3}$$

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \sum_{p=1}^P Q_{ijkt} c_{ijktp} u_{ijktp} \leq B \tag{5.8-4}$$

$$Q_{ijkt} \leq C \tag{5.8-5}$$

$$w_{ijkt} = \begin{cases} 1 & \text{if } Q_{ijkt} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.8-6}$$

$$\sum_{p=1}^P u_{ijktp} = \begin{cases} 1 & \text{if } Q_{ijkt} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.8-7}$$

$$Q_{ijkt} \in \mathbb{R}, x_{ijkt} \geq 0; w_{ijkt}, u_{ijktp} \in \text{binary}; g(y_{1k}, y_{2k}) \geq 0;$$

$$(\text{for } i = 1, 2, \dots, I; j = 1, 2, \dots, J; k = 1, 2, \dots, K; t = 1, 2, \dots, T)$$

In Equation (5.8), there are several constraints that can increase the complexity of the model. The restriction (5.8-1) obtains the initial inventory of each buyer in each period remained from the previous period. Equation (5.8-2) determines that each vendor's warehouse has a limited capacity. Also, each vendor has a limited production capacity that is shown by Equation (5.8-3). The total available budget to purchase the products is restricted where the relevant limitation is proposed by Equation (5.8-4). Furthermore, Equation (5.8-5) shows an upper bound (due to the production limitations) for the order quantities. Equation (5.8-6) represents a binary

variable for making an order and Equation (5.8-7) says each buyer must purchase each item maximum at a price break point in each time.

5.3 Solution methodologies

In the current section, in order to solve the proposed two-echelon supply chain model, a MPSO is applied where a GA algorithm is used to compare and evaluate the performance of the proposed algorithm. The MPSO is explained as the following stages.

5.3.1 Initializing the parameters

Firstly, the parameters concerned with the MPSO including the number of particles (NoP), the number of generations (NoG) and two parameters λ_1 and λ_2 , are defined. Additionally, the position and velocity, which are two variables in PSO algorithm, are initialized using Equations (5.9) and (5.10), respectively. In Equation (5.9), z_0^l is the initial position of particle l ($l=1,2,...,NoP$), z_{\min} is the lower and upper bound on the design variables (here, $z_{\min}=0$ and $z_{\max}=C$ for Q) and $rand$ is a random number in the interval $(0,1)$. Also, in Equation (5.10) v_0^l is the initial velocity of the particle l ($l=1,2,...,NoP$) and η is the constant time increment and assumed 1. These parameters are also adjusted using the Taguchi method which is explained in the next section. Figure 5.3 shows a representation of the particles, where the values of the order quantity are generated randomly in the interval of $[0, C]$ and the number of particles in each generation is set to NoG . Moreover, the vendors and customers are assumed to be located into a certain region with coordinates $y_1 \in [0,100]$ and $y_2 \in [0,100]$.

$$z_0^l = z_{\min} + rand(z_{\max} - z_{\min}) \quad (5.9)$$

$$v_0^l = \frac{z_0^l}{\eta} \quad (5.10)$$

Q_{1111}	Q_{1112}	...	Q_{111T-1}	Q_{211T-1}	...	Q_{ijkt}	...	Q_{IJKT-1}	y_{11}	...	y_{1K}	...	y_{2K}
------------	------------	-----	--------------	--------------	-----	------------	-----	--------------	----------	-----	----------	-----	----------

Figure 5.3: The presentation of a particle

5.3.2 Evaluating the particles

In this stage, we evaluate each of the particles by using Equation (5.8). Figure 5.4 represents a population of the particles since the objective value of the particles is depicted by TC .

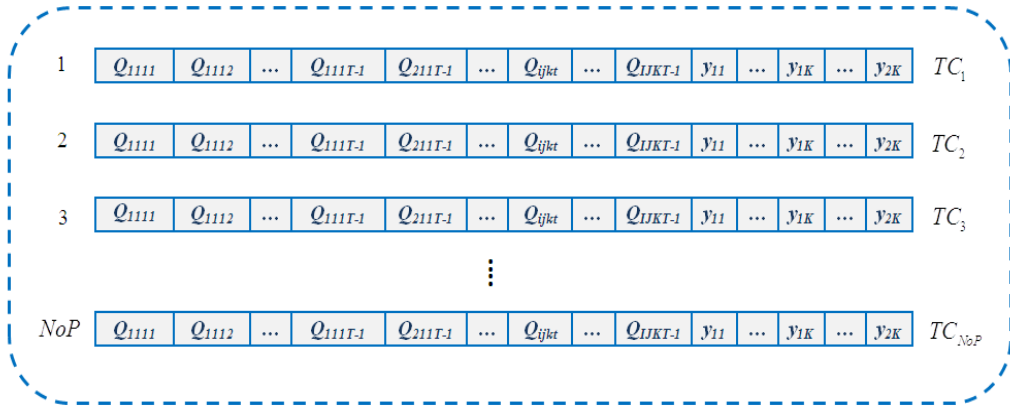


Figure 5.4: The representation of a population of the particles

5.3.3 Updating the velocities and positions

In order to search the solutions in the feasible area of the problem, the velocity and position of the particles in each generation of PSO are updated as the following formulas.

$$v_{n+1}^l = w.v_n^l + \lambda_1.r_1.(pBest_n^l - z_n^l) + \lambda_2.r_2.(gBest_n - z_n^l) \quad (5.11)$$

$$z_{n+1}^l = z_n^l + \eta.v_{n+1}^l \quad (5.12)$$

In Equation (5.11), r_1 and r_2 are two numbers generated randomly in the interval of (0,1), the coefficients λ_1 and λ_2 are the given acceleration constants towards $pBest$ and $gBest$, respectively, and W is the inertia weight where is expressed as Equation (5.13) (Naka et al., 2001). Furthermore, $pBest_n^l$ and $gbest_n$ are the best fitness value for particle l until time n , ($n=1,2,...,NoG$) and the best particle among all until time n , respectively.

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})}{NoG} \cdot n \quad (5.13)$$

In Equation (5.13), NoG is the maximum number of iterations and n is the current number of iteration. (Shi & Eberhart, 1999) and (Naka et al., 2001) have claimed the best result will be obtained since. $[w_{\min}, w_{\max}] = [0.4, 0.9]$

An important aspect of generation and initializing the particles is that solutions must be feasible and satisfy the constraints. A penalty function approach is used for those particles that do not satisfy all the constraints given in Equation (5.14).

$$R(x) \leq L \quad (5.14)$$

Therefore, the corresponding penalty function is defined as follows:

$$F(x) = \begin{cases} 0 & \text{if inequality is satisfied} \\ (R(x) - L)^\alpha & \text{otherwise} \end{cases} \quad (5.15)$$

where α is the coefficient of the penalty function (here, $\alpha=10$).

5.3.4 Stopping criteria

In a meta-heuristic algorithm, the stopping criterion can be reached by specifying CPU time, a specific value of the objective value, or a specified number of

generation. In this research, the number of generation (NoG) has been adopted to stop the optimization process.

Furthermore, in order to validate the performance of MPSO, a GA is applied based on the following steps.

- I. Initialize the chromosomes, the number of generation (NoG), the number of population (NoP), the probability of crossover (P_c) and the probability of mutation (P_m).
- II. Evaluate the chromosomes by using Equation (5.8).
- III. Select the chromosomes based on the tournament method to enter the production pool (each time select two chromosomes and one with the best objective value is selected).
- IV. Perform crossover operator on the chromosomes. First, for each chromosome, generate a random number between 0 and 1. Those are selected for the crossover operator that their related random numbers are set to less or equal to P_c . Next, two chromosomes out of the selected chromosomes are chosen for the crossover operator randomly. If $chro_1$ and $chro_2$ are the two chosen chromosomes (parents), the offsprings are generated as:

$$\begin{aligned} off_1 &= rand.chro_1 + (1 - rand).chro_2 \\ off_2 &= (1 - rand).chro_1 + rand.chro_2 \end{aligned} \quad (5.16)$$

where $rand$ is a random number between 0 and 1 and off_1 and off_2 are the offspring.

- V. Perform mutation operator. In this operator, a random number is also generated between 0 and 1 for each chromosome for which the 1s are

selected for mutation operator that have values less or equal to P_m . Hence, the mutation operator generates the new chromosomes from the selected chromosomes as follows: in each chromosome a variable is selected randomly and is changed in the range randomly.

- VI. Perform elitism operator. Those chromosomes that are not selected for both crossover and mutation operators enter directly to next generation in the order of their objective values while the number of population reaches NoP .
- VII. Stop the algorithm based on is reaching a specific number of generation.

5.4 Data generating, Parameter setting and computational results

In this section, first a range of random numerical examples is generated to evaluate the algorithms on the supply chain model. Secondly, we design a Taguchi method to tune the parameters of the algorithms where MINITAB software version 15 is used to analyze the data. Finally, a case study is considered to evaluate the model on a real world case. In order to solve the model proposed by Equation (5.8), MATLAB (R2013a) software is used to code the algorithms on a PC with RAM 4GH and CPU 2.5 dual cores.

5.4.1 Generating data

The examples were constructed by generating values for the parameters provided in this study. Parameter generation was summarized in the list shown in Table 5.1. Table 5.1 depicts the value ranges used to generate the different problems which are classified in three categories of small, medium and large. In other words, the value ranges of all parameters are shown in Table 5.1. For example according to Table 5.1, (20 to 50) means the minimum demand values for all items that each customer can order in all periods (seasons) are equal to 20 and the maximum demand values

would be 50, respectively. For each generated problem shown in Tables 5.5, 5.6 and 5.7, the demand values for each item in each period are variable while these values can be changed for each item in different periods (seasons) like those explained in Table 4.3. Table 5.1 uses the strategy applied in (Dayarian et al., 2016; Diewert et al., 2009; Mogale et al., 2017; Saracoglu et al., 2014; Tanksale & Jha, 2017) to generate data randomly. In addition, according to (Costantino et al., 2016), demands in seasonal items can be random.

Table 5.1: The input data of the test problems

d_{ijkt}	T_{ijkt}	h_{ijkt}	A_{ijkt}	C	α_i	c_{ijktp}
(20 to 50)	(1 to 3)	(1 to 20)	(1 to 20)	(1 to 150)	(1 to 100)	(10 to 20)

A very common approach to generate data is to generate a wide range of numerical examples with different sizes randomly to show how the algorithms perform on different problems with different sizes (Dayarian et al., 2016; Diewert et al., 2009; Saracoglu et al., 2014; Tanksale & Jha, 2017). In this study, three categories based on size each one with 10 instances were generated randomly. The Small-scale instances were generated with 5 to 10 buyers, 1 to 5 products and vendors and 1 to 3 periods. The Medium-scale instances were generated with 11 to 20 buyers, 6 to 10 products, 1 to 10 vendors, and 1 to 5 periods since in Large-scale instances these values are 20 to 30 for buyers, 11 to 15 for products and vendors and 6 to 10 for periods. Additionally, the number of price-break point for all the three categories is considered to be 4.

Table 5.2: Sizes of the proposed instances

Description	Buyers (I)	Products (J)	Vendors (K)	Periods (T)	Price-break point
Small-scale	[5-10]	[1-5]	[1-5]	[1-3]	4
Medium-scale	[11-20]	[6-10]	[1-10]	[1-5]	4
Large-scale	[20-30]	[11-15]	[11-15]	[6-10]	4

5.4.2 Parameter setting

One of the major problems in using meta-heuristic algorithms is that the algorithm parameters can take different values for different problems. A searchable space refers to the possibility of measuring the distance (similarity) between any two candidates so that a sensible search space (landscape) could be defined. In other words, for any candidate, it should be possible to find out which candidates are close to it and which candidates are far from it (Yuan & Gallagher, 2005). In this work, in order to reduce the computational time to obtain the best solution, the proposed MPSO and GA algorithms were tuned using the Taguchi method. The Taguchi method is a fractional factorial experiment introduced by Taguchi applied as an efficient alternative for full factorial experiments (Shavandi et al., 2012). The Taguchi method is also one of the most well-known approaches that is utilized for tuning the meta-heuristic parameters used in the literature recently (Mousavi et al., 2013; Mousavi et al., 2013; Mousavi & Niaki, 2012; Mousavi et al., 2013; Peace, 1993; Sadeghi et al., 2013b). As aforementioned, MPSO and GA were applied to find the optimal solutions of the two-echelon supply chain network in Equation (5.8) at which λ_1 , λ_2 , NoP and NoG are the input parameters of MPSO and P_c , P_m , NoP and NoG are the input parameters of GA. In this research, the “Smaller is Better” type of response has been employed (since the goal is to minimize S/N), where S/N is given as.

$$S / N_{ratio} = -10 \times \log\left(\frac{\sum_{l=1}^{\beta} Y_l}{\beta}\right) \quad (5.17)$$

In Equation (5.17), Y and β (here, $\beta = 1$) are the response and the number of orthogonal arrays, respectively. To design the Taguchi for both meta-heuristic

algorithm parameters, we used L_9 design where the values and levels of the parameters are given in Table 5.3. The values in Table 5.3 were obtained after numerous tests and analyses on the current instances of the categories using the frequent runs of the algorithms. We represented the experimental design for Problem No. 1 of Small-scale category in details in order to show how the parameters were tuned in each of instances. Table 5.4 represents the orthogonal arrays along with their responses for both MPSO and GA for Problem Number 1 (*No. 1*) of Small-scale shown in Table 5.5. In Table 5.4, *A* and *B* show the factors of λ_1 and λ_2 in MPSO and P_C and P_m in GA respectively since *C* and *D* are equivalent to *NoP* and *NoG* in both MPSO and GA. Moreover, the sixth and seventh columns of Table 5.4 represent the responses of MPSO and GA approaches for Problem *No. 1* of Small-scale category respectively. Figures 5.5 and 5.6 display the mean S/N ratio plot of the MPSO and GA for Problem No. 1 of Small-scale respectively. According to Figures 5.5 and 5.6, the optimal levels of the MPSO's factors are $\lambda_1=2.5, \lambda_2=2$, *NoP*=20 and *NoG*=100 where these levels for GA's factors are $P_C=0.6, P_m=0.2$, *NoP*=40 and *NoG*=500 for GA.

Table 5.3: The MPSO and GA parameters' levels

<i>Algorithm</i>	<i>Parameters</i>	<i>1</i>	<i>2</i>	<i>3</i>
<i>MPSO</i>	λ_1	1.5	2	2.5
	λ_2	1.5	2	2.5
	<i>NoP</i>	20	30	40
	<i>NoG</i>	100	200	500
<i>GA</i>	P_C	0.5	0.6	0.7
	P_m	0.08	0.1	0.2
	<i>NoP</i>	30	40	50
	<i>NoG</i>	200	300	500

Table 5.4: The experimental results on the MPSO and GA parameters for Problem No. 1 of Small-scale

Exp. No.	A	B	C	D	MPSO	GA
1	1	1	1	1	30456	31416
2	1	2	2	2	28953	29356
3	1	3	3	3	28091	32431
4	2	1	2	3	31371	33253
5	2	2	3	1	32646	31098
6	2	3	1	2	31467	31829
7	3	1	3	2	30633	29809
8	3	2	1	3	34968	30963
9	3	3	2	1	34894	32646

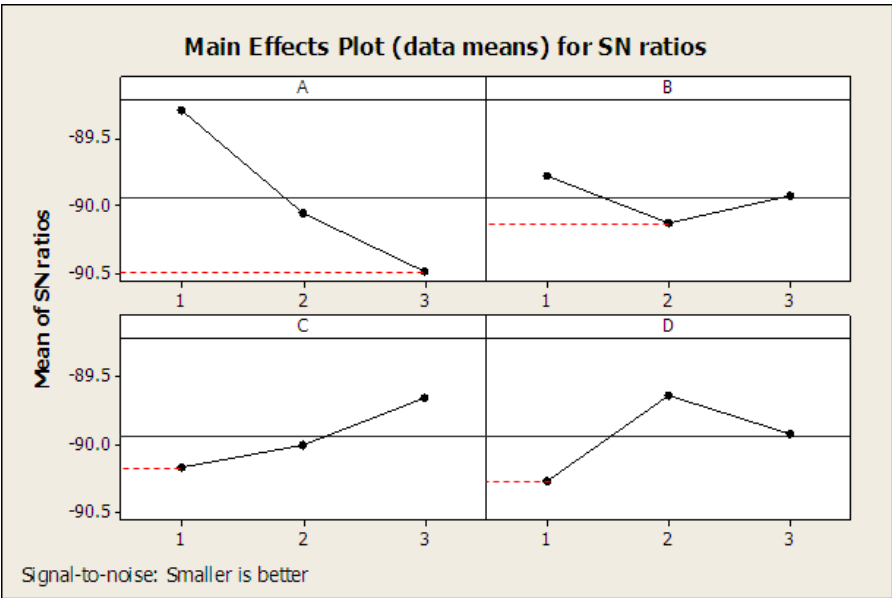


Figure 5.5: The mean S/N ratio plot of the MPSO on Problem No. 1 of Small-scale

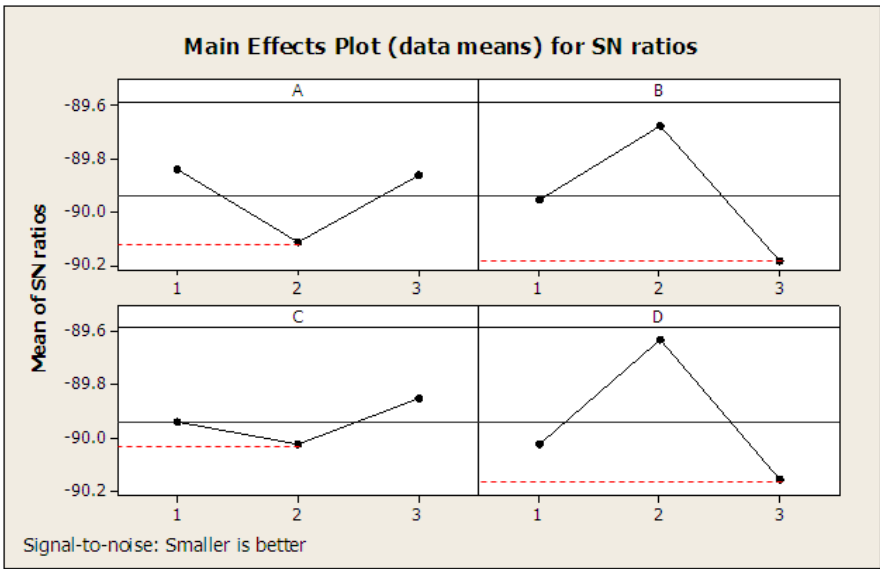


Figure 5.6: The mean S/N ratio plot of the GA on Problem No. 1 of Small-scale

5.4.3 The results and comparisons

In this section, we compare the results obtained from both MPSO and GA with each other on the 10 instances generated from the three aforementioned categories to find the best methodology for solving the proposed two-echelon supply chain model. Tables 5.5, 5.6 and 5.7 demonstrate the input parameter and the objective values of both MPSO and GA for each one of the instances of the three categories generated in the range given in Table 5.2. In these Tables, the optimal values of the algorithm parameters are obtained using the Taguchi method with L_9 design. Furthermore, the optimal values of the objective function for MPSO and GA (which are TC) resulted from each instance of the three categories are also shown in the columns 14 and 19 of Tables 5.5 to 5.7, respectively.

In order to compare the performance of MPSO and GA in terms of the objective function, several approaches were employed in this research. First, we have taken the average and standard deviation (St. Dev) of each 10 instances for all the category problems showing in the last two rows of Tables 5.5, 5.6 and 5.7. The results of average and St. Dev of the instances in the three categories demonstrate that MPSO has outperformed GA.

A graphical approach shown in Figure 5.7 is also secondly applied to compare the performance of the algorithms on the 10 generated instances of Small-scale, Medium-scale and Large-scale categories. According to Figure 5.7, in each three (a) Small-scale, (b) Medium-scale and (c) Large-scale, the proposed MPSO seems to have a better efficiency than GA.

Table 5.5: The input parameters and the objective function of MPSO and GA for Small-scale problems

Problem No.	Buyer	product	vendor	Period	Price-break point	S_j	B	f_k	MPSO					GA				
									λ_1	λ_2	NoP	NoG	TC	P_C	P_m	NoP	NoG	TC
1	2	2	2	2	4	2500	18000	500	2.5	2	20	100	26367	0.6	0.2	40	500	28936
2	3	3	3	3	4	19000	125000	1500	2.5	2.5	20	200	434250	0.7	0.1	40	300	457312
3	5	2	3	2	4	4000	75000	900	2	2.5	30	100	222371	0.7	0.1	30	300	247460
4	5	3	3	3	4	15000	210000	2500	2.5	2.5	30	200	824325	0.7	0.08	40	300	886870
5	5	2	2	2	4	3000	55000	700	2	2.5	20	200	142930	0.6	0.1	30	200	165640
6	5	5	5	2	4	16000	290000	1900	2	2	30	500	1129720	0.7	0.2	50	500	1383900
7	8	2	2	2	4	3500	70000	1400	2.5	2.5	30	200	348390	0.6	0.1	30	300	453462
8	8	3	3	3	4	17000	350000	3700	2	2.5	20	100	1159504	0.6	0.1	30	200	1392806
9	10	2	3	3	4	15000	270000	3500	2	2.5	30	100	1229134	0.7	0.1	30	200	1421100
10	10	4	5	2	4	16000	450000	3200	1.5	2	30	200	1897425	0.6	0.2	50	300	1958302
Average	-	-	-	-	-	-	-	-	-	-	-	-	741441.6	-	-	-	-	839578.8
St. Dev	-	-	-	-	-	-	-	-	-	-	-	-	605014.8	-	-	-	-	662608.1

Table 5.6: The input parameters and the objective function of MPSO and GA for Medium-scale problems

Problem No.	Buyer	product	vendor	Period	Price-break point	s_i	B	f_k	<i>MPSO</i>					<i>GA</i>				
									λ_1	λ_2	<i>NoP</i>	<i>NoG</i>	<i>TC</i>	P_C	P_m	<i>NoP</i>	<i>NoG</i>	<i>TC</i>
1	12	6	6	4	4	120000	4000000	21000	1.5	2	30	200	15747123	0.7	0.1	30	300	16817032
2	15	6	6	4	4	130000	4050000	22000	2	2	40	500	20956050	0.6	0.08	50	500	21646300
3	15	6	6	5	4	200000	5400000	30000	2	2.5	50	500	36222230	0.7	0.1	40	500	36649110
4	15	6	10	4	4	210000	6800000	22500	2.5	1.5	40	500	36885490	0.6	0.2	40	300	40413290
5	20	6	2	2	4	9000	580000	9600	2	2.5	30	200	2775400	0.7	0.2	30	300	2977320
6	20	6	3	3	4	35000	1850000	20000	2.5	2.5	20	100	10260430	0.7	0.2	30	200	10774120
7	20	7	2	2	4	110000	720000	11000	2	1.5	20	100	3126310	0.6	0.1	30	300	3499230
8	20	7	3	3	4	46000	2100000	23000	1.5	2.5	40	200	12290540	0.6	0.2	40	300	13117650
9	18	8	2	2	4	17500	715000	12200	2.5	2.5	30	500	3249270	0.5	0.2	50	500	3570419
10	18	8	3	4	4	100000	3300000	36500	2.5	2	40	100	14889710	0.7	0.08	30	300	14964700
Average	-	-	-	-	-	-	-	-	-	-	-	-	15640255	-	-	-	-	16442917
St. Dev	-	-	-	-	-	-	-	-	-	-	-	-	12559251	-	-	-	-	13207845

Table 5.7: The input parameters and the objective function of MPSO and GA for Large-scale problems

No. Problem	Buyer	product	vendor	period	Price-break point	s_i	B	f_k	<i>MPSO</i>					<i>GA</i>				
									λ_1	λ_2	<i>NoP</i>	<i>NoG</i>	<i>TC</i>	P_c	P_m	<i>NoP</i>	<i>NoG</i>	<i>TC</i>
1	21	11	11	6	4	170000	8000000	45000	1.5	1.5	30	500	27456901	0.6	0.1	20	300	30234321
2	22	11	11	6	4	170000	8000000	45000	2.5	2	30	200	28092034	0.7	0.08	30	500	31092340
3	25	12	11	6	4	180000	8500000	48000	2.5	1.5	50	100	31003290	0.7	0.2	50	300	33785005
4	25	12	12	6	4	205000	880000	50000	2	2	30	200	31980245	0.5	0.1	30	300	34098210
5	27	11	11	6	4	180000	850000	45000	2.5	2	30	200	30900700	0.6	0.2	40	200	33109230
6	28	12	12	6	4	200000	880000	47500	2	1.5	50	500	32980134	0.7	0.08	30	300	34561200
7	29	11	11	7	4	160000	820000	44000	2.5	2.5	30	500	31003410	0.7	0.2	40	300	34789123
8	30	11	11	6	4	170000	810000	46000	2	2	50	100	29891360	0.6	0.1	40	200	33421876
9	30	12	12	6	4	210000	890000	50000	2.5	2	40	200	33209130	0.7	0.2	50	300	36490080
10	30	15	15	10	4	450000	1100000	75000	2	2	30	500	48710900	0.6	0.2	50	500	52082400
Average	-	-	-	-	-	-	-	-	-	-	-	-	3466347.5	-	-	-	-	3567944.6
St. Dev	-	-	-	-	-	-	-	-	-	-	-	-	5986070	-	-	-	-	6139669.1

Finally, to compare the performance of the algorithms, we have performed an independent two-sample t-test with a 95% confidence for the instances of the categories where a hypothesis test for means of MPSO and GA when their standard deviations are unknown is as:

$$H_0 : \mu_{MPSO} - \mu_{GA} = \delta_0 \text{ versus } H_1 : \mu_{MPSO} - \mu_{GA} \neq \delta_0 \quad (5.18)$$

at which μ_{MPSO} and μ_{GA} are the means of the objective values of the two algorithms and δ_0 is the hypothesized difference between the means of the algorithms. Tables 5.8, 5.9 and 5.10 depict the results of t-test for the instances of Small-scale, Medium-scale and Large-scale categories respectively. From Tables 5.8 and 5.9, it is clear that P-values are greater than commonly chosen 0.05-levels. Hence, there is no evidence for a difference in the performance of the algorithms on the instances of Small-scale and Medium-scale categories. However, the P-value in Table 5.10 also shows that there is a no difference between the two algorithms for the instances of Large-scale category. Therefore, MPSO has performed efficiently for solving the integrated location allocation two-echelon supply chain problem.

For more understanding of the solutions, the optimal order quantity and location of the vendors obtained by MPSO for Problem No. 1 of Small-scale category are shown in Figure 5.8. Furthermore, in order to clarify the trend of the solutions obtained from the first generation to the last, Figures 5.9 and 5.10 demonstrate the convergence path of the objective values for the MPSO and GA for Problem No. 10 of Medium-scale category, respectively. Figure 5.11 also shows a representation of the optimal locations of the vendors among the buyers obtained by MPSO for Problem No. 7 of Medium-scale category.

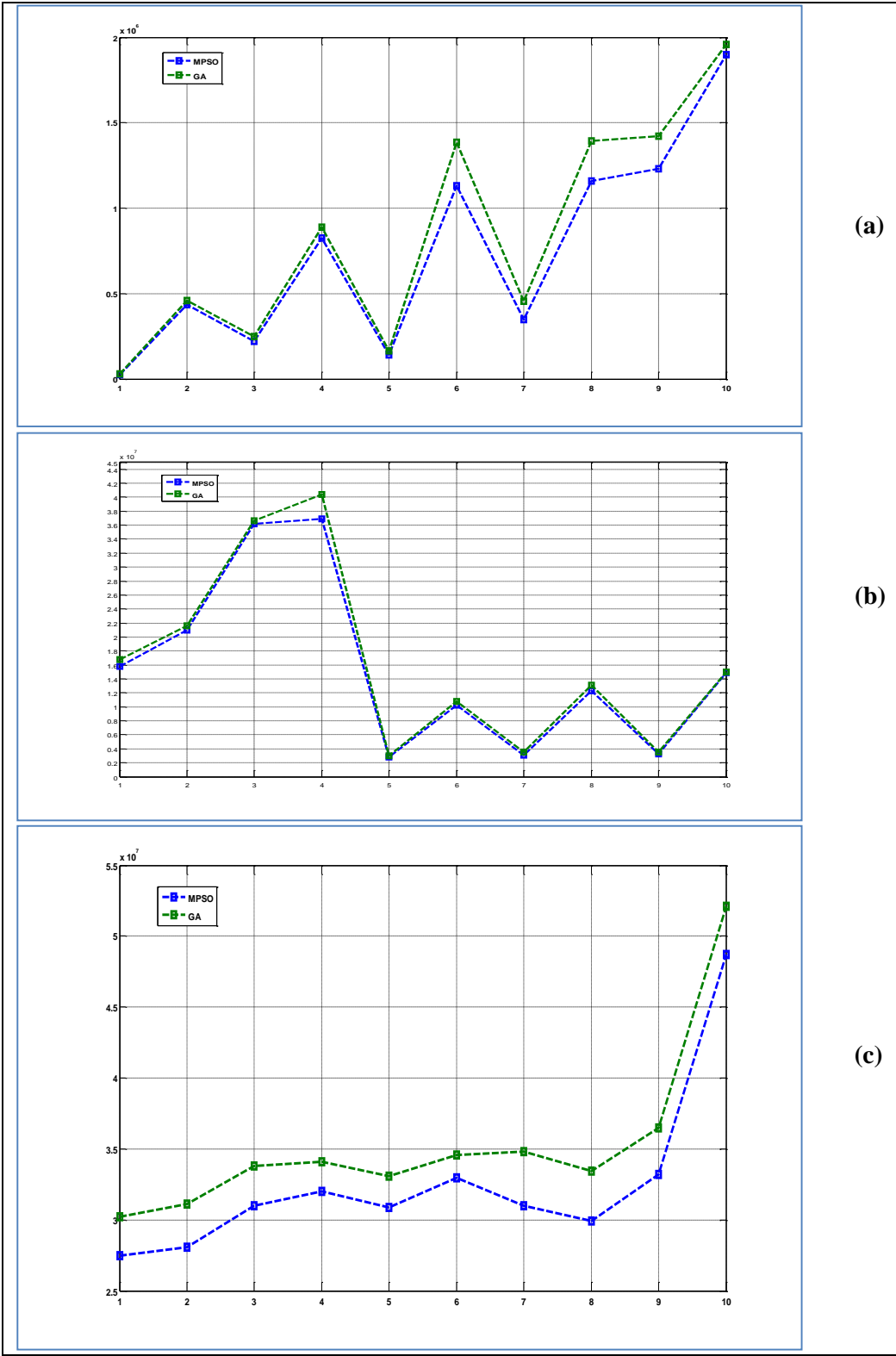


Figure 5.7: The graphical representation of the objective function resulted from MPSO and GA on the generated instances of (a) Small-scale, (b) Medium-scale and (c) Large-scale

Table 5.8: The ANOVA for the instances of Small-scale category

Methodology	N	Mean	St-Dev	SE-Mean	T-Value	P-Value
PSO	10	741442	605015	191322	-0.35	0.734
GA	10	839579	662608	209535	-	-

Table 5.9: The ANOVA for the instances of Medium-scale category

Methodology	N	Mean	St-Dev	SE-Mean	T-Value	P-Value
PSO	10	15640255	12559251	3971584	-0.14	0.891
GA	10	16442917	13207845	4176687	-	-

Table 5.10: The ANOVA for the instances of Large-scale category

Methodology	N	Mean	St-Dev	SE-Mean	T-Value	P-Value
PSO	10	32522810	5986070	1892962	-1.05	0.309
GA	10	35366379	6139669	1941534	-	-

Q_{1111}	Q_{2111}	Q_{1211}	Q_{2211}	Q_{1121}	Q_{2121}	Q_{1221}	Q_{2221}	Y_{11}	Y_{21}	Y_{12}	Y_{22}
100	5	45	25	17	61	48	30	24.17	3.38	76.34	14.28

Figure 5.8: A representation of the optimal solution obtained by MPSO for Problem No. 1 of Small-scale category

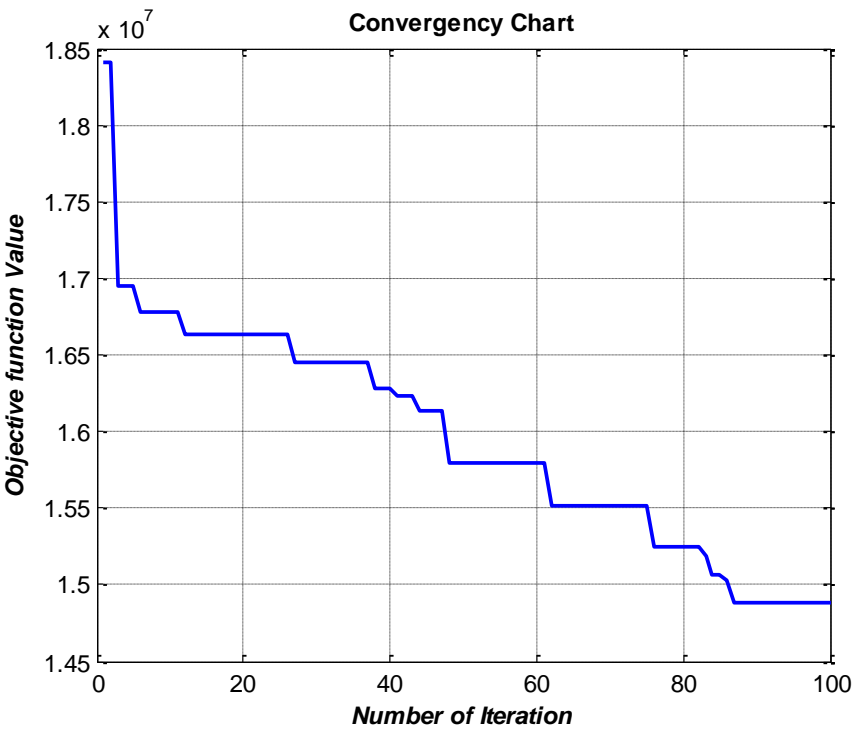


Figure 5.9: The convergence path of MPSO for Problem No. 10 of Medium-scale category

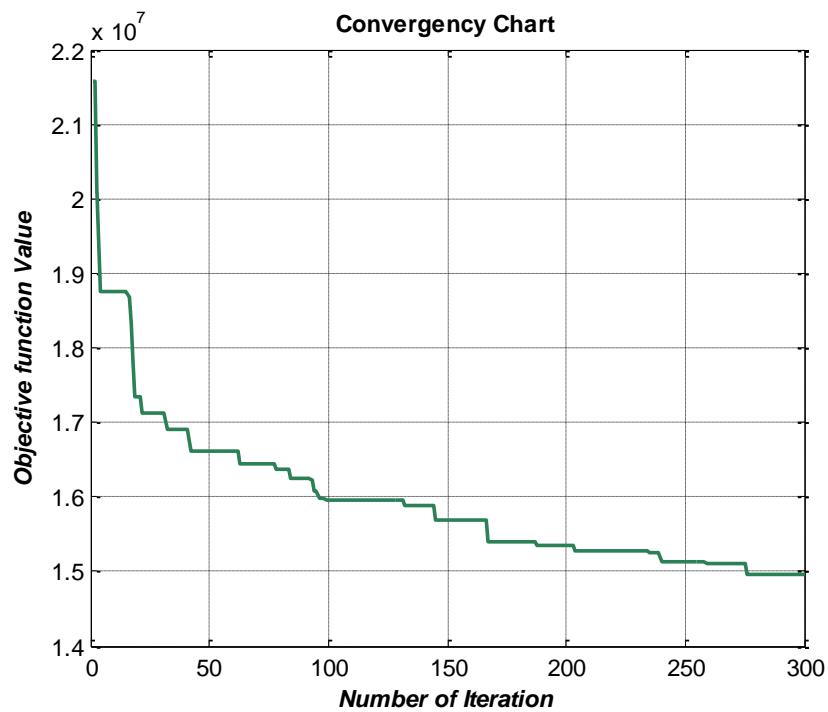


Figure 5.10: The convergence path of GA for Problem No. 10 of Medium-scale category

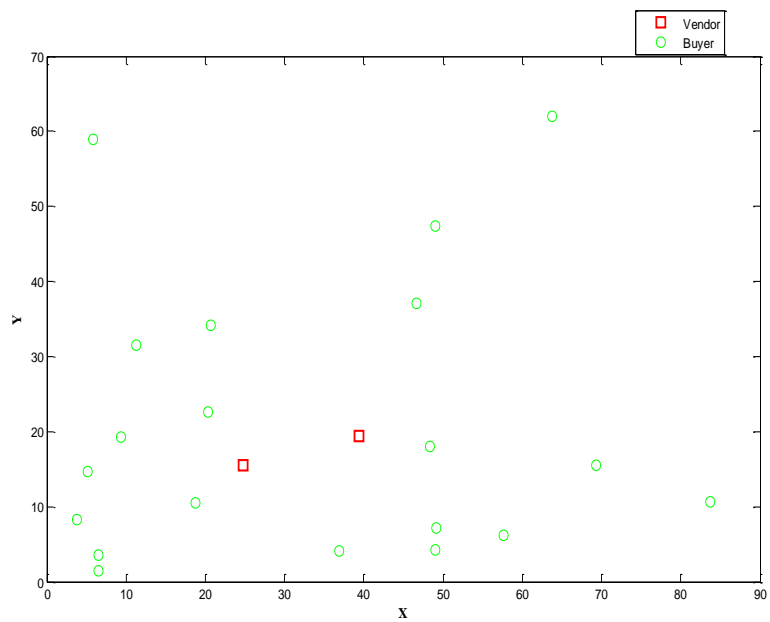


Figure 5.11: The optimal location of the vendors among the buyers obtained by MPSO for problem No. 7 of Medium-scale category

5.4.4 A Case Study

The wood industry manufacturing-trade High Gloss, Melamine and finished products which is located in Iran was used to collect real data for evaluating the

problem. The company imports a number of products as well as MDF, HMA, HGF and PVC from the countries of Malaysia, Turkey, Thailand, China and Germany every three months. There are three main buyers located in the cities of Tehran, Kerman and Shiraz purchasing the items from the company as well where their demand rates are almost deterministic and known. Furthermore, the planning horizon is considered to be one year (12 months) including four periods (months), where the replenishment process is taking place every three months. The warehouse spaces and the purchasing costs of all the buyers were the same. The company aims to locate two vendors in the potential locations so that the total inventory costs are minimized. In addition, based on the data proposed by the company which is shown in Appendix B the area of the study is supposed to be a square area with coordinates distributed in the interval $[0, 100]$ in 10 Km where the buyers and the vendors are connected to each other with the paths.

Tables 5.11 to 5.13 show the data collected from the first, second and third buyers located in Tehran, Kerman and Shiraz respectively which are presented in Appendix B. The locations coordinates of the buyers are proposed by the company to be (22, 19), (18, 68) and (59, 65) in 10 Km. Figure 5.12 shows the coordinates of the buyers' locations placed in Tehran, Kerman and Shiraz in Iran which are proposed by the marketing team department of the company (ORAMAN) where Mahshahr city is assigned to be placed in the location with coordinates (0, 0).



Figure 5.12: The coordinates of the buyers' locations proposed by the marketing department of ORAMAN Wood Industry shown on Google Maps (Appendix B)

The values of the other parameters are given as follows: $C = 600$, $f_k = 18000$ (the production capacity of each vendor $k = 1, 2$), $S_k = 5600m^2$, and $B = 192,000,000$ IT. The price of each item is proposed under AUD policy which is the same for all vendors as follows:

$$P_{MDF} = \begin{cases} 53000 & 0 < Q \leq 100 \\ 52000 & Q > 100 \end{cases}$$

$$P_{HMA} = 30000$$

$$P_{HGF} = \begin{cases} 810000 & 0 < Q \leq 70 \\ 800000 & Q > 70 \end{cases}$$

$$P_{PVC} = \begin{cases} 160000 & 0 < Q \leq 70 \\ 150000 & Q > 70 \end{cases}$$

Furthermore, the ordering costs and holding costs for all the items are as the same proposed in Table 4.9.

Table 5.11: The data for a real problem collected from the first buyer located in Tehran

Product	d_{1j11}	d_{1j12}	d_{1j13}	d_{1j21}	d_{1j22}	d_{1j23}	A_{1j1t}	A_{1j2t}	h_{1jkt}	$s_{1j1t}(m^2)$	$s_{1j2t}(m^2)$
<i>MDF (sheet)</i>	110	110	120	100	100	110	2000	2000	1000	250	250
<i>Adhesive (Kg)</i>	90	100	90	70	40	110	200	200	100	100	100
<i>HGF (120m)</i>	90	85	80	90	55	110	1000	1000	1000	300	300
<i>PVC (120m)</i>	70	70	50	65	70	100	300	300	1000	400	400

Table 5.12: The data for a real problem collected from the second buyer located in Kerman

Product	d_{2j11}	d_{2j12}	d_{2j13}	d_{2j21}	d_{2j22}	d_{2j23}	A_{2j1t}	A_{2j2t}	h_{2jkt}	$s_{2j1t}(m^2)$	$s_{2j2t}(m^2)$
<i>MDF (sheet)</i>	100	100	170	100	110	130	2100	2000	1100	250	250
<i>Adhesive (Kg)</i>	50	95	80	80	70	65	180	200	100	100	100
<i>HGF (120m)</i>	60	70	70	65	80	80	1000	1100	900	300	300
<i>PVC (120m)</i>	60	80	70	70	75	95	320	300	900	400	400

Table 5.13: The data for a real problem collected from the third buyer located in Shiraz

Product	d_{3j11}	d_{3j12}	d_{3j13}	d_{3j21}	d_{3j22}	d_{3j23}	A_{3j1t}	A_{3j2t}	h_{1jkt}	$s_{3j1t}(m^2)$	$s_{3j2t}(m^2)$
<i>MDF (sheet)</i>	100	90	105	110	80	105	2200	2200	1200	250	250
<i>Adhesive (Kg)</i>	100	105	90	80	110	110	200	100	100	100	100
<i>HGF (120m)</i>	115	105	100	100	105	120	1000	1200	1100	300	300
<i>PVC (120m)</i>	80	90	80	85	90	115	300	300	1000	400	400

The total optimal supply chain cost, the optimal locations of the vendors and the optimal order quantities obtained by MPSO were obtained as follows, where the CPU time elapsed by MPSO was 36.506.

$TC_{MPSO} = 393,637,000$ IT includes:

Transportation costs (TrC) = 137,700,000 IT

Holding cost (HC) = 67,500,000 IT

Purchasing cost (PC) = 187,800,000 IT

$$y_1 = (32, 44) ; y_2 = (42, 25)$$

$$Q_{1MDFkt} = [490, 520, 460, 490, 450, 530]$$

$$Q_{1HMAkt} = [520, 480, 450, 490, 530, 440]$$

$$Q_{1HGFkt} = [480, 500, 460, 440, 530, 480]$$

$$Q_{1PVCKt} = [460, 440, 480, 500, 470, 460]$$

$$Q_{2MDFkt} = [520, 480, 460, 460, 470, 510]$$

$$Q_{2HMAkt} = [460, 520, 440, 450, 440, 470]$$

$$Q_{2HGFkt} = [490, 500, 460, 530, 480, 470]$$

$$Q_{2PVCKt} = [450, 480, 490, 460, 500, 470]$$

$$Q_{3MDFkt} = [450, 500, 450, 490, 470, 500]$$

$$Q_{3HMAkt} = [500, 450, 510, 520, 520, 450]$$

$$Q_{3HGFkt} = [510, 440, 440, 470, 530, 480]$$

$$Q_{3PVCKt} = [480, 520, 530, 520, 450, 490]$$

The total optimal supply chain cost, the optimal locations of the vendors and the optimal order quantities obtained by GA were computed as follows, where the CPU time elapsed by GA was 44.832.

$Tc_{GA} = 490,730,000$ IT includes:

Transportation costs (TrC) = 193,220,000 IT

Holding cost (HC) = 107,040,000 IT

Purchasing cost (PC) = 190,470,000 IT

$y_1 = (31, 59) ; y_2 = (40, 37)$

$Q_{1MDFkt} = [450, 530, 480, 480, 470, 520]$

$Q_{1HMAkt} = [490, 490, 510, 470, 510, 480]$

$Q_{1HGFkt} = [460, 470, 490, 510, 530, 470]$

$Q_{1PVCkt} = [480, 460, 520, 510, 480, 450]$

$Q_{2MDFkt} = [500, 510, 480, 450, 520, 510]$

$Q_{2HMAkt} = [450, 490, 480, 520, 500, 480]$

$Q_{2HGFkt} = [510, 530, 480, 480, 490, 480]$

$Q_{2PVCkt} = [480, 480, 470, 470, 520, 500]$

$Q_{3MDFkt} = [510, 520, 440, 480, 480, 490]$

$Q_{3HMAkt} = [530, 470, 500, 510, 490, 490]$

$Q_{3HGFkt} = [510, 470, 480, 460, 510, 490]$

$Q_{3PVCkt} = [480, 500, 470, 530, 520, 510]$

Figures 5.13 and 5.14 show a representation of the optimal locations of the vendors among the buyers obtained by MPSO for the proposed case study.

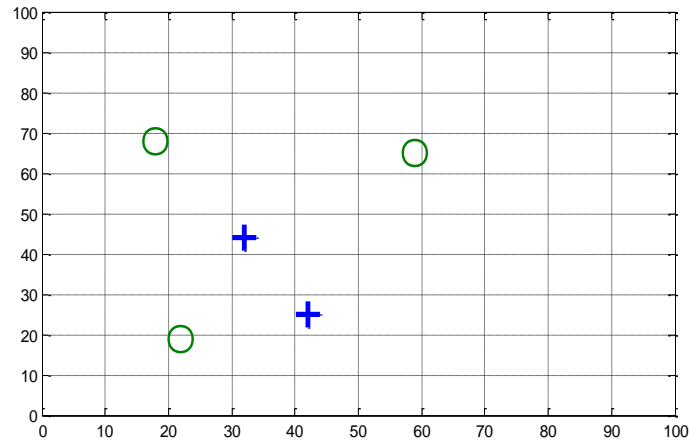


Figure 5.13: The optimal location of the vendors among the buyers obtained by MPSO for the case study

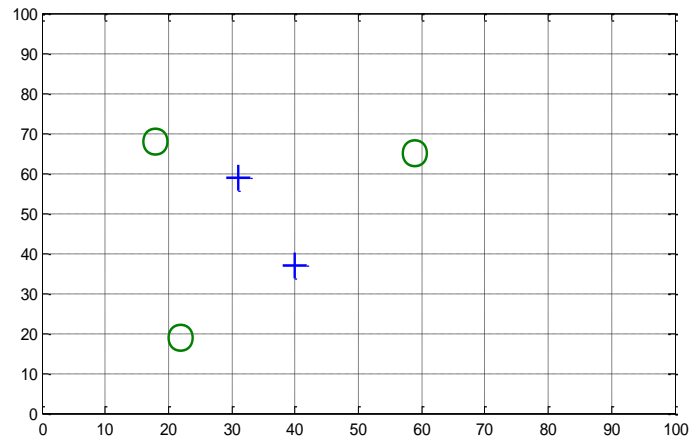


Figure 5.14: The optimal location of the vendors among the buyers obtained by GA for the case study

5.5 Summary

In this chapter, a two-echelon supply chain network for a inventory control problem was investigated where the vendors stored the produced products into their warehouses. The retailers made the orders for these products under all unit discount policy. The main goal of the problem was to find the optimal order quantity of the products purchased by the buyers in addition to determining the optimal locations of the vendors among the known location of buyers so that the total supply chain

cost comprising transportation, holding and purchasing costs is minimized. The distances between the buyers and the vendors were supposed as the Euclidean distance. To solve the proposed supply chain model, a MPSO algorithm was employed where a GA was utilized to validate the results of the proposed algorithm. Taguchi method was also applied to set the parameters of the two algorithms. The results of the algorithms showed the MPSO has a better performance than the GA in terms of the objective function on both the generated instances of the three categories and the case study.

CHAPTER 6: OPTIMIZING A LOCATION ALLOCATION INVENTORY PROBLEM IN A TWO-ECHELON SUPPLY CHAIN NETWORK: A MODIFIED FRUIT FLY OPTIMIZATION ALGORITHM

6.1 Introduction

In this current Chapter, a mixed binary-integer nonlinear mathematical model is developed for a location-allocation two-echelon retailer-distributor supply chain problem in which a variety of the products are offered by the distributors to retailers. The distributors manufacture the products and store them in their own warehouses with limited capacities where there are certain places to hold each item. The retailers order different products from different distributors at specific time-periods based on their requirements. The products are delivered in certain packets with specific number of items using trucks with limited capacities. The planning horizon of the problem comprises multiple periods, where the replenishment process is taken place at the beginning of these periods. Here, different products in different periods may face shortages as a combination of lost sales and backorders. Moreover, it is possible the whole quantities of the products cannot be sold in different periods. Therefore, a number of items remain in the warehouses. In case of a shortage for a product in a period, the retailer should make an order at least as much as the demand becomes satisfied. Besides, due to some uncertain constraints, the distributors are not able to produce the products more than a specific value and also the total available budget is limited. The distributors provide the products to the retailers under all-unit and incremental quantity discount policies. The main goal is to find the optimal locations of distributors among the retailers and to determine the order quantities of the products ordered by the retailers from the distributors in different periods so that the total supply chain costs are minimized. Figure 6.1 depicts the supply chain network proposed in this work. Figure 6.2 shows a graphical illustration of the replenishment

process of the inventory system in the proposed supply chain network. In Figure 6.2, it is clear that in some periods there are items (inventory) available (the trapezoids above the line) and there are also shortages in some periods (the trapezoids below the line).

This chapter is an extension of the previous work presented in Chapter 5 where the problem has been improved in some concerns significantly. In this model, the shortages are allowed for some items where some customers will stay to receive their ordered items (a fraction β_{ijkt} of customers are backorder) and some other customers will go to another company to get their orders (a fraction $(1-\beta_{ijkt})$ of customers are lost sale). Therefore, the shortage costs are added to the total system costs. In Chapter 5, shortages are not allowed. To encourage customers to buy more items, two discount policies as well as AUD for some items and IQD for other items are proposed by distributors where in the previous model only AUD policy was given. In addition, the items are delivered in the special boxes with pre-specific number of items where the items are not sent using the special boxes in the model of Chapter 5. Moreover, the proposed model is formulated and solved for two Euclidean and Square Euclidean distance functions among the distributors and retailers separately where the prior model is formulated only with Euclidean distance. Finally, for the first time in the literature FOA is improved for solving an inventory-supply chain and location allocation problem where the FOA showed better performance than other algorithms in terms of objective function and CPU time graphically and statistically while PSO was the best in Chapter 5.

The rest of the chapter is organized as follows. In Section 6.2, the two-echelon retailer–distributor supply chain problem is defined. Also, the problem was formulated in this section. The solution algorithms come in Section 6.3 where a

modified fruit fly optimization algorithm (MFOA) was proposed. Some numerical examples are generated in Section 6.4 where the Taguchi approach is applied to tune the parameters of the algorithms. Furthermore, a case study is considered in Section 6.4. Finally, the conclusion is settled in Section 6.5.

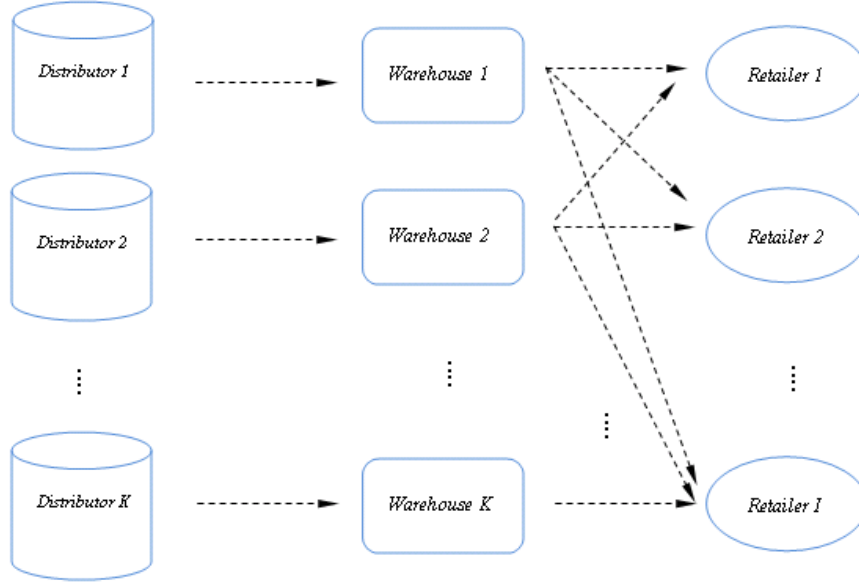


Figure 6.1: The proposed supply chain network

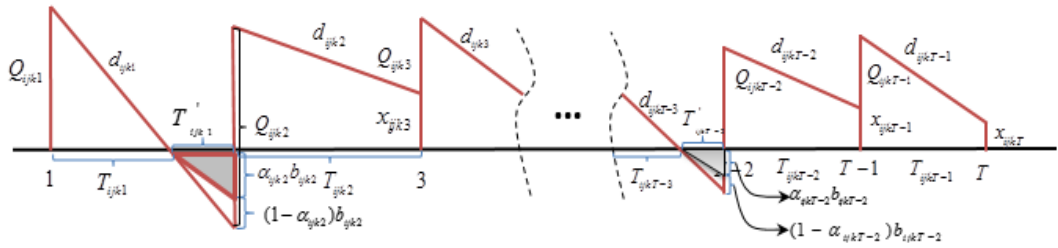


Figure 6.2: The graphical illustration of the inventory system in the proposed supply chain network

6.2 Modeling the proposed problem

To model the problem, let first introduce the assumptions of the supply chain model. Then, the parameters and the variables of the model are defined next.

The assumptions involved in the supply chain problem at hand are:

- i. Replenishment is instantaneous, i.e. the delivery time is assumed negligible.
Based on other inventory research works, this assumption is made for the sake of simplicity and will not impose considerable influence on the modeling aspects.
- ii. The retailers demand rates of all products are independent of one another and are fixed in a period.
- iii. Retailers receive all products from the distributors in the certain packets with specific capacities, which are the ordered quantity of products is delivered in packets of a fixed-sized batch.
- iv. In case of shortage, a fraction of demand is considered backorder and a fraction as lost sales.
- v. The initial order quantity of all products is considered zero.
- vi. The planning horizon, a future time-period during which departments that support production plan production work and determine material requirements, is finite and known. In the planning horizon, there are periods of an equal length.
- vii. The retailers are located in a certain region.
- viii. The distance between the retailers and distributors are assumed to be either Euclidean or square Euclidean.

6.2.1 Indices, parameters, variables, and decision variables

In order to formulate the problem, the indices, the parameters, the variables, and the decision variables were defined as:

(a) *Indices:*

$i = 1, 2, \dots, I$ An index for the retailers

$j = 1, 2, \dots, J$ An index for the products

$k = 1, 2, \dots, K$	An index for the distributors
$t, t = 0, 1, \dots, T$	An index for the periods
$m, m = 1, 2, \dots, M$	An index for the price break points

(b) Parameters and variables:

λ_{ijktm} :	A binary variable that is set to 1 if product j is purchased by retailer i from distributor k at price break point m in period t ; 0 otherwise
D_{ijkt} :	The demand of retailer i for the j^{th} product from distributor k in period t
T_{ijkt} :	Total time elapsed up to and including the order of j^{th} product by retailer i from distributor k in the t^{th} period
T'_{ijkt} :	The time length of period t at which the demand of product j purchased by retailer i from distributor k reaches zero
h_{ijkt} :	Inventory holding cost per unit time of the j^{th} product produced by distributor k for retailer i in period t
f_k :	The production capacity of distributor k
n_{ijkt} :	The fixed batch size of the j^{th} product ordered by retailer i from distributor k in period t
O_{ijkt} :	Transportation cost per unit of the j^{th} product carried out from distributor k to retailer i in period t
c_{ijktm} :	The cost paid by retailer i to purchase one unit of the j^{th} product from distributor k at m^{th} price break point in period t

- q_{ijktm} : m^{th} price break-point offered by distributor k to retailer i for the j^{th} product in period t (here, $q_{ijkt1} = 0$)
- s_{ijkt} : The required warehouse space of retailer i to store one unit of the j^{th} product purchased from distributor k in period t
- s_k : Total available space of the storage belongs to distributor k
- $a_i = (a_{1i}, a_{2i})$: The coordinates of the location of retailer i
- $P(x_{1k}, x_{2k})$: The potential region of distributor k
- $d(x_k, a_i)$: A function indicating the distance between the locations of distributor k and retailer i
- B : The total available budget of the supply chain network
- ub : An upper bound for the available number of boxes
- v_{ijkt} : A binary variable set to 1 if retailer i purchases product j from distributor k in period t ; 0 otherwise
- TC : The total supply chain costs
- μ_{ijkt} : Backorder cost per unit of the j^{th} product ordered by retailer i to distributor k in period t
- μ'_{ijkt} : Lost sale cost per unit of the j^{th} product ordered by retailer i to distributor k in period t
- β_{ijkt} : The percentage of backorder demand of the j^{th} product ordered by retailer i to distributor k in period t

(c) Decision variables:

$x_k = (x_{1k}, x_{2k})$: The coordinates of the location of distributor k

- R_{ijkt} : Number of packets of the j^{th} product ordered by retailer i to distributor k in period t
- b_{ijkt} : Shortage quantity of the j^{th} product ordered by retailer i to distributor k in period t
- Q_{ijkt} : Ordering quantity of the j^{th} product ordered by retailer i to distributor k in period t
- y_{ijkt} : The initial positive inventory of the j^{th} product ordered by retailer i to distributor k in period t (in $t=1$, the beginning inventory of all products is zero)

6.2.2 The supply chain cost

The objective function of the proposed model is minimizing the total supply chain costs including transportation, holding, shortage, and purchase costs. The transportation cost, TrC , that includes the ordering cost as well was formulated as:

$$TrC = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T Q_{ijkt} O_{ijkt} v_{ijkt} d(x_k, a_i) \quad (6.1)$$

where the function $d(x_k, a_i)$ is either the Euclidean or the square Euclidean distance shown in Equations (6.2) and (6.3), respectively.

$$d(x_k, a_i) = \sqrt{(x_{1k} - a_{1i})^2 + (x_{2k} - a_{2i})^2} \quad (6.2)$$

$$d(x_k, a_i) = (x_{1k} - a_{1i})^2 + (x_{2k} - a_{2i})^2 \quad (6.3)$$

According to Figure 6.2, the holding cost, HoC , was obtained as

$$HoC = \sum_i \sum_j \sum_k \sum_{t=1}^{T-1} \{ (y_{ijkt} + Q_{ijkt} + y_{ijkt+1})(T_{ijkt} + T'_{ijkt}) \} h_{ijkt} / 2 \quad (6.4)$$

where it is calculated using the sum of the trapezoidal areas of the positive order quantities in Figure 6.2. Moreover, the shortage cost ShC , as a combination of backorders and lost sales costs, was derived using the sum of the trapezoidal areas of the negatives as

$$ShC = \sum_i^I \sum_j^J \sum_k^K \sum_t^{T-1} \left\{ \frac{\mu_{ijkt} b_{ijkt}}{2} (\beta_{ijkt} T'_{ijkt}) + \frac{\mu'_{ijkt} b_{ijkt}}{2} ((1 - \beta_{ijkt}) T'_{ijkt}) \right\} \quad (6.5)$$

To formulate the purchasing cost under the all-unit discount policy (AUD), let the price break points be considered as

$$\begin{cases} c_{ijkt1} & q_{ijkt1} \leq Q_{ijkt} < q_{ijkt2} \\ c_{ijkt2} & q_{ijkt2} \leq Q_{ijkt} < q_{ijkt3} \\ & \vdots \\ c_{ijktM} & q_{ijktM} \leq Q_{ijkt} \end{cases} \quad (6.6)$$

Then, the total purchasing cost under the AUD policy (TpA) is given by

$$TpA = \sum_i^I \sum_j^J \sum_k^K \sum_t^{T-1} \sum_{m=1}^M Q_{ijkt} c_{ijktm} \lambda_{ijktm} \quad (6.7)$$

In the incremental quantity discount policy, the purchasing cost offered by distributor k to retailer i for each unit of the j^{th} product in period t depends on its ordered quantity, where in each price discount-point, it is obtained by

$$\begin{cases} c_{ijkt1} & q_{ijkt1} \leq Q_{ijkt} < q_{ijkt2} \\ c_{ijkt1} q_{ijkt1} + (Q_{ijkt} - q_{ijkt2}) & q_{ijkt2} \leq Q_{ijkt} < q_{ijkt3} \\ & \vdots \\ c_{ijkt1} q_{ijkt1} + c_{ijkt2} q_{ijkt2} + \dots + c_{ijktM} (Q_{ijkt} - q_{ijktM}) & q_{ijktP} \leq Q_{ijkt} \end{cases} \quad (6.8)$$

Therefore, the total purchasing cost under this policy (TpI) is

$$TpI = \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \left\{ \sum_{m=1}^{M-1} ((q_{ijktm+1} - q_{ijktm}) c_{ijktm} \lambda_{ijktm}) + (Q_{ijkt} - q_{ijktM}) c_{ijktM} \lambda_{ijktM} \right\} \quad (6.9)$$

Therefore, with the assumption that the total purchasing cost is $Pc = TpA + TpI$, the total supply chain cost, considered the fitness value thereafter, is obtained by

$$TC = TrC + HoC + ShC + Pc \quad (6.10)$$

6.2.3 The constraints

Several constraints are presented based on the assumptions made in Section 6.2. First, the inventory of product j ordered by retailer i from distributor k in period t , which are y_{ijkt} , can be either positive denoted by y_{ijkt} , or negative denoted by b_{ijkt}

(the shortage quantity in period t). In other words, $y_{ijkt} = \begin{cases} 0 & \text{if } y_{ijkt} < 0 \\ y_{ijkt} & \text{otherwise} \end{cases}$ and

$b_{ijkt} = \begin{cases} -y_{ijkt} & \text{if } y_{ijkt} < 0 \\ 0 & \text{otherwise} \end{cases}$. Furthermore, the beginning inventory of product j

ordered by retailer i to distributor k in period $t+1$ is equal to its beginning inventory in the previous period t plus the ordered quantity minus a coefficient of the demand that is sold. Or

$$y_{ijkt+1} = y_{ijkt} + Q_{ijkt} - D_{ijkt}(T_{ijkt} + T'_{ijkt}) \quad (6.11)$$

As the ordered quantity of product j by retailer i from distributor k in period t , Q_{ijkt} , is delivered in R_{ijkt} packets, each containing n_{ijkt} products, the next constraint is

$$Q_{ijkt} = n_{ijkt} R_{ijkt} \quad (6.12)$$

Moreover, the number of available packets to deliver product j to retailer i by distributor k in period t is limited, we have,

$$R_{ijkt} \leq ub \quad (6.13)$$

The total budget to buy the products is limited to B . Hence

$$TpA + TpI \leq B \quad (6.14)$$

The warehouse space of the distributor k for products is limited to S_k . Thus

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{t=1}^T (Q_{ijkt} + y_{ijkt}) s_{ijkt} \leq S_k \quad (6.15)$$

Besides, due to some production limitations the distributors were not able to produce the products more than a certain quantity, which are

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{t=1}^T Q_{ijkt} \leq f_k \quad (6.16)$$

Finally, as at most one order can be placed by each retailer to each distributor for a product in a period, and that the product can be purchased at one price break point, we have

$$\sum_{p=1}^P \lambda_{ijkt} = \begin{cases} 1 & \text{if } Q_{ijkt} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.17)$$

Therefore, the complete mathematical model of the supply chain problem is:

$$\begin{aligned} MinTC = & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^T Q_{ijkt} O_{ijkt} v_{ijkt} d(x_k, a_i) + \\ & \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \{ (y_{ijkt} + Q_{ijkt} + y_{ijkt+1})(T_{ijkt} + T'_{ijkt}) \} h_{ijkt} / 2 \\ & + \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \left\{ \frac{\mu_{ijkt} b_{ijkt}}{2} (\beta_{ijkt} T'_{ijkt}) + \frac{\mu'_{ijkt} b_{ijkt}}{2} ((1 - \beta_{ijkt}) T'_{ijkt}) \right\} + \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \sum_{m=1}^M Q_{ijkt} c_{ijktm} \lambda_{ijktm} \\ & + \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \left\{ \sum_{m=1}^{M-1} ((q_{ijktm+1} - q_{ijktm}) c_{ijktm} \lambda_{ijktm}) + (Q_{ijkt} - q_{ijktM}) c_{ijktM} \lambda_{ijktM} \right\} \end{aligned}$$

Subject to:

$$y_{ijkt+1} = y_{ijkt} + Q_{ijkt} - D_{ijkt} (T_{ijkt} + T'_{ijkt}) \quad (6.18)$$

$$\sum_{i=1}^I \sum_{k=1}^K \sum_{t=1}^T (Q_{ijkt} + y_{ijkt}) s_{ijkt} \leq S_k$$

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{t=1}^T Q_{ijkt} \leq f_k$$

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K \sum_{t=1}^{T-1} \sum_{p=1}^P Q_{ijkt} c_{ijktp} \lambda_{ijktp} \leq B$$

$$R_{ijkt} \leq ub$$

$$Q_{ijkt} > b_{ijkt}$$

$$Q_{ijkt} = n_{ijkt} R_{ijkt}$$

$$v_{ijkt} = \begin{cases} 1 & \text{if } Q_{ijkt} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{p=1}^P \lambda_{ijktp} = \begin{cases} 1 & \text{if } Q_{ijkt} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ijkt} = \begin{cases} 0 & \text{if } y_{ijkt} < 0 \\ y_{ijkt} & \text{otherwise} \end{cases}$$

$$b_{ijkt} = \begin{cases} -y_{ijkt} & \text{if } y_{ijkt} < 0 \\ 0 & \text{otherwise} \end{cases}$$

$$T'_{ijkt} = \frac{b_{ijkt+1}}{d_{ijkt}}$$

$$Q_{ijkt}, y_{ijkt} \geq 0; w_{ijkt}, u_{ijktp}, Z_{ijkt} \in \text{binary}; (\text{for } i = 1, \dots, I; \text{for } j = 1, 2, \dots, J; \\ \text{for } k = 1, 2, \dots, K; \text{for } t = 1, 2, \dots, T; \text{for } m = 1, 2, \dots, M); P(x_{1k}, x_{2k}) \geq 0;$$

$$b_{ijk1} = 0; y_{ijk1} = 0; Q_{ijkT} = 0;$$

While the shortage values b_{ijk1} are decision variables, based on the proposed formula in Equation (6.18), i.e. $y_{ijk1} = y_{ijkt} + Q_{ijkt} - D_{ijkt}(T_{ijkt} + T'_{ijkt})$, when the value $y_{ijkt} < 0$ we have shortage for item j in period t otherwise we don't have shortage (see $b_{ijkt} = \begin{cases} -y_{ijkt} & \text{if } y_{ijkt} < 0 \\ 0 & \text{otherwise} \end{cases}$). Furthermore, β_{ijkt} is the fraction of the shortages

considered to be backorder (i.e. $\beta_{ijkt} b_{ijkt}$) and $(1 - \beta_{ijkt}) b_{ijkt}$ is considered to be lost sale.

In other words, when the inventory level of an item in a period i.e. y_{ijkt} is negative it means that a shortage happens.

6.3 Solving methodologies

The model derived in (6.18) is a mixed binary-integer mathematical formulation that is hard to solve using an analytical approach. Hence, a modified fruit fly optimization algorithm (MFOA) is proposed in this section for solution. In addition, as there is no benchmark available in the literature to validate the results obtained, two other meta-heuristics called particle swarm optimization (PSO) and simulated annealing (SA) are utilized as well.

The procedure of the original fruit fly optimization algorithm is summarized as follows:

- i. Initialize parameters, including maximum number of generations and population size.
- ii. Initialize a population of fruit fly groups randomly
- iii. Construct several fruit flies randomly around the fruit fly group using osphresis for the foraging to generate a population
- iv. Evaluate all the flies of the population to obtain the smell concentration values (fitness value) of each fruit fly
- v. Find the best fruit fly with the maximum smell concentration value using vision for the foraging, and then let the fruit fly group fly towards the best one.
- vi. End the algorithm if the maximum number of generations is reached.

The modified fruit fly optimization of this research is expressed in the following steps.

6.3.1 Initializing and representing the flies

In the MFOA, each fly is a solution of the problem shown in Figure 6.3. This solution consists of the number of packets for each product ordered by each retailer from each distributor in each period i.e. R_{ijkt} along with to the location coordinate of the distributors i.e. x_{ki} . The number of packets are generated uniformly in the range $[0, ub]$. The location coordinates of the distributors in the area $P(x_{1k}, x_{2k})$ are also uniformly generated with $x_{1k} \in [0, 100]$ and $x_{2k} \in [0, 100]$. The population of the fruit flies (group) is initialized as shown in Figure 6.4.

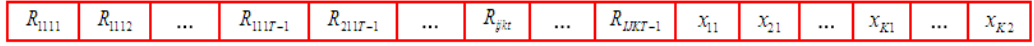


Figure 6.3: The representation of a fruit fly

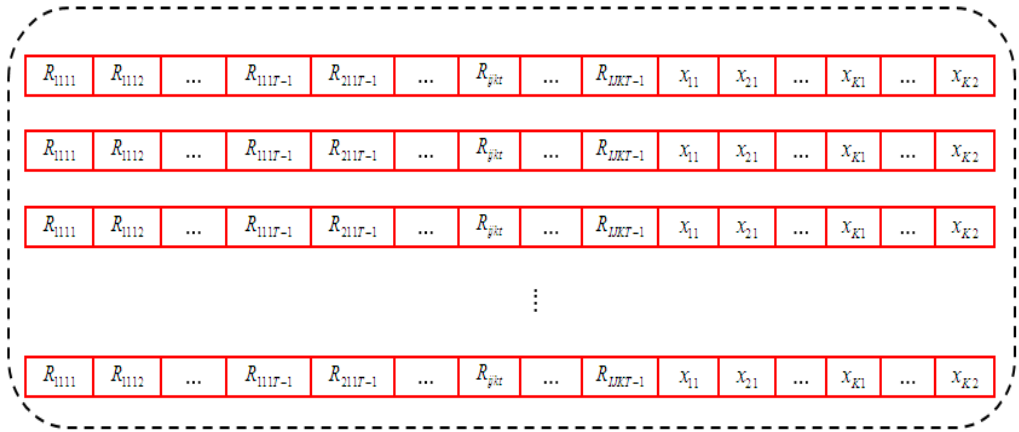


Figure 6.4: The representation of a population of the flies

6.3.2 Smell-based and vision-based search

In the smell-based search process, first NP fruit flies were generated around each fruit fly group to construct a sub-population using the position and velocity of the flies in PSO algorithm. The positions of the flies were obtained using

$$X_{e+1}^{ld} = X_e^{ld} + v_{e+1}^{ld} \quad (6.19)$$

where X_e^{ld} is the position of l^{th} fly in dimension d and iteration e , $l = 1$ to NS , $d = 1, 2, \dots, D$, v_{e+1}^{ld} is its velocity, and NS is the number of fillies. Moreover, the velocities of the flies used in Equation (6.19) are calculated based on the equation derived in Equation (6.20).

$$v_{e+1}^{ld} = w.v_e^{ld} + \mu_1.Rand.(pB_e^{ld} - X_e^{ld}) + \mu_2.Rand.(gB_e^d - X_e^d) \quad (6.20)$$

In Equation (6.20), w is the inertia weight to control the magnitude of the old velocity v_e^{ld} in calculation of the new velocity v_{e+1}^{ld} , pB_e^{ld} and gB_e^d are the position of the best local and the best global fly respectively, μ_1 and μ_2 determine the significance of pB_e and gB_e^d and $Rand$ is a uniformly distributed real random number between 0 and 1.

After the positions were updated in Equation (6.19), the neighborhood search technique make sure that the searching takes place within the boundary areas defined by the inequality constraints given in (6.18). In this regard, often, the newly generated individuals may not satisfy the capacity constraints. To guarantee the feasibility of the individuals, a penalty function shown in Equation (6.21) was employed to penalize infeasible solutions.

$$F(x) = \begin{cases} 0 & \text{if inequality is satisfied} \\ (R(x) - L)^\alpha & \text{otherwise} \end{cases} \quad (6.21)$$

In Equation (6.21), α is the severity of the penalty function (here, $\alpha = 10$) and $R(x)$ and L are referred to a typical constraint $R(x) \leq L$.

6.3.3 Global vision-based search

In this search, the algorithm finds the best fly among the best flies found in sub-populations in terms of their fitness values.

6.3.4 Stopping criterion

While several stopping criteria such as reaching a specific CPU time, converging to a specific value of the fitness function, or having a specific number of generations are common to stop a meta-heuristic, the third criterion is used to stop the algorithm in NG generations.

6.4 Testing and comparisons

In this section, two kinds of problems are tested on the problem to evaluate the algorithms as well.

6.4.1 The generated data

Thirty numerical examples of three-sizes, small, medium, and large are randomly generated in this section to assess the performance of the MFOA. In small-size problems, the number of retailers were randomly generated in the range from 2 to 5, where the number of products, distributors, periods, and price break-points all were generated in (1 to 5). In the medium-size problems, the range for the number of retailers was (5 to 10), for products is (6 to 9), for distributors was (5 to 9), for periods was (6 to 10), and for the number of price-break points was (5 to 6). These ranges for the large-size problems are (10 to 50), (9 to 30), (9 to 20), (9 to 50), and (5 to 6), respectively.

To validate the results obtained using MFOA and to evaluate its performance in terms of the fitness function, two other meta-heuristics are utilized to solve the problems. All algorithms have been coded on MATLAB R2013a and the codes have been executed on a computer with 3.80 GHz and 4 GB of RAM. In these codes, both the Euclidean and the square Euclidean distances are used.

Tables 6.1 to 6.3 contain general data of the 20 problem instances along with the fitness functions resulted using the three algorithms with Euclidean and square Euclidean distances on small-size, medium-size, and large-size problems, respectively. Moreover, the parameters of all algorithms are tuned by the Taguchi method. To be more specific, consider *Prob. No. 11* of the medium-size category with Euclidean and square Euclidean distances. The number of parameters is 5 in MFOA (μ_1 , μ_2 , NS , NP , and NG), 4 in PSO (C_1 , C_2 , NP , and NG) , and 3 in SA algorithms ($Alpha$ (the cooling schedule), $inlo$ (internal loop), and NG). The parameters of the algorithms and their levels for this problem are shown in Tables 6.4 and 6.5, respectively.

The L_{27} array of the Taguchi method has been used to tune the parameters of MFOA while the L_9 array was used to calibrate the parameters of the other two algorithms. For this problem, the optimal levels of the parameters of the three algorithms have been given in Tables 6.6 and 6.7 for Euclidean and square Euclidean, respectively. In order to determine the optimal levels of the parameters of the algorithms of this study, Taguchi used the “smaller is better” case in its (S/N) formula. Moreover, Figures 6.5 and 6.6 show the mean signal to noise ratio (S/N) plot of each level of the factors for *Prob. No. 11* of medium-size problems for the MFOA, PSO, and SA algorithms with Euclidean and square Euclidean distances, respectively.

To evaluate and to compare the performance of the MFOA with the ones of PSO and SA algorithms, two graphical and statistical approaches were used. Figures 6.7 and 6.8 display the graphical representation of the fitness values for the three category problems with Euclidean and square Euclidean distances, respectively. The algorithms have different performance on different problems, which are not based

on a specific pattern, but it is based on the ability of each algorithm to solve each problem with different size. Because of this, the three algorithms have different fitness values and CPU time on different problems with different sizes for both Euclidean and Square Euclidean functions. For example in Problem No. 18 of Table 6.1, MFOA algorithm has an impressive performance than other algorithms in terms of fitness value for Square Euclidean distance while the results are near each other in other problems. In other words, MFOA has been found to be a powerful algorithm than others in terms of fitness value and CPU time for solving a problem with five retailers, three types of product, three distributors, three periods and also three price-break points. Moreover, the results of this algorithm are better but near the results obtained by PSO and SA on other problems in terms of fitness value and CPU time. The detailed costs obtained by three algorithms on Problem No. 18 of Table 6.1 for Square Euclidean distance are as follows:

The results obtained by MFOA are: $TC = 47730$ including $TrC = 28763$, $HoC = 8157$, $ShC = 6750$ and $Pc = 4060$.

The results obtained by PSO are: $TC = 58762$ including $TrC = 33322$, $HoC = 10436$, $ShC = 10034$ and $Pc = 4970$.

The results obtained by SA are: $TC = 59134$ including $TrC = 34734$, $HoC = 10168$, $ShC = 9192$ and $Pc = 5040$.

To compare the results obtained by the algorithms for Problem No. 18 of Table 6.1, there is a considerable gap between the total costs obtained by MFOA and those optimized by PSO and SA as much as 11032 and 11404, respectively. The main concern of this gap comes from Transportation Cost (TrC) where TrC obtained by MFOA, is 4559 less than TrC obtained by PSO and 5971 less than the one obtained by SA. Therefore, the transportation cost of items considerably affects the

performance of MFOA directly to be better than SA and PSO. Furthermore, shortage cost (ShC) obtained by MFOA is also calculated to be 3284 less than PSO and 2442 less than SA which is the second more effective cost on the performance of MFOA. There is also a gap of 2279 between MFOA and PSO and a gap of 2011 between MFOA and SA in terms of holding cost (HoC). There is a very low gap between the purchasing cost (Pc) obtained by MFOA and both PSO and SA the gap is less than 1000. As a result, the TC performance of MFOA on Problem No. 18 of Table 6.1 is directly affected by three parameters TrC, HoC and ShC while the effect of Pc on the MFOA performance is not too much.

According to these figures, it can be concluded that MFOA has better performance than both PSO and SA in terms of fitness value. Figures 6.9 and 6.10 that represent the box plots of the fitness values for the 30 problem instances of the three categories with Euclidean and square Euclidean distances can also affirm this conclusion, where it seems that MFOA has the best efficiency in comparison with the two other algorithms in terms of fitness value.

However, based on the analysis of variance (ANOVA) results shown in Tables 6.8 to 6.13 for all the problem instances using both the Euclidean and square Euclidean distances, the hypothesis on the equality of algorithms' means, which are $H_0 : \mu_{MFOA} = \mu_{PSO} = \mu_{SA}$ cannot be rejected at 95% confidence level in favor of the alternative $H_1 : \mu_{MFOA} \neq \mu_{PSO} \neq \mu_{SA}$. In other words, the three algorithms do not have differences in terms of fitness values in all the problem instances of different sizes. This validate the results obtained using MFOA. Note that based on the results in Tables 6.14 to 6.16 that show the CPU times taken by each algorithm to solve small, medium, and large-size problems, respectively, the MFOA has the faster algorithm to solve the problems in all the three problem categories.

Table 6.1: General data for small-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances

Prob. No.	I	J	K	T	M	ub	B	f_k	S_k	SA		PSO		MFOA	
										EUD	SEUD	EUD	SEUD	EUD	SEUD
1	1	2	2	2	2	50	1000	450	550	15014	18139	14551	17982	14489	17854
2	1	3	2	3	2	50	1200	500	700	16781	20170	15980	19332	15321	18715
3	1	3	3	3	3	50	1500	550	800	18329	23412	17142	22892	17901	22329
4	1	4	3	4	3	50	1700	610	850	19488	24937	19120	24200	18842	23849
5	1	4	4	4	4	50	1800	750	900	21579	27331	20325	26879	19753	26056
6	2	2	2	2	2	50	2100	840	1000	23560	28620	23114	28201	22705	27800
7	2	3	3	3	3	50	2400	890	1200	24773	29781	24142	29119	23650	28689
8	2	4	4	3	3	50	2600	1050	1400	26890	32711	26367	32319	25617	31614
9	2	4	4	4	4	50	2700	1150	1500	30142	34720	29761	34429	29130	33860
10	3	2	3	2	3	50	3000	1450	1650	33672	38721	33249	38317	32741	37872
11	3	3	3	3	3	70	3400	1500	1800	36412	40543	36030	39654	35653	39160
12	3	4	3	4	3	70	3800	1600	1900	37509	43094	37205	42720	36548	42106
13	3	2	3	4	3	70	4200	1750	2100	39540	46832	39209	46320	38651	45764
14	3	4	4	4	4	70	4500	1830	2300	42319	50830	41750	49430	40983	48880
15	4	3	3	2	2	70	4400	1800	2300	43138	51560	42798	51107	42257	50794
16	4	4	4	4	4	70	5000	2000	2500	45721	55391	45340	54970	44803	54572
17	5	2	3	2	3	70	4600	1900	2450	46728	57321	46410	57003	45807	56672
18	5	3	3	3	3	70	5100	2050	2600	48542	59134	48116	58762	47608	47703
19	5	4	4	4	4	70	5800	2200	2600	51478	63098	51101	62780	50632	62195
20	5	5	5	5	5	70	6700	2500	2900	55730	68921	55202	68430	54872	67832

Table 6.2: Some general data for medium-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances

Prob. No.	I	J	K	T	M	ub	B	f_k	S_k	SA		PSO		MFOA	
										EUD	SEUD	EUD	SEUD	EUD	SEUD
1	5	6	5	6	5	50	7500	2600	3000	58430	69356	57890	68832	57210	68167
2	5	6	6	6	6	50	7700	2800	3200	60326	72317	59698	71873	59120	71254
3	6	2	2	4	4	50	7000	2400	2800	57980	68432	57458	67943	56812	67218
4	6	3	3	3	3	50	7200	2600	3100	59725	69319	59209	68801	58725	68193
5	6	4	4	4	4	50	7500	2700	3300	63428	72140	62872	71659	62046	70899
6	6	5	5	5	5	50	7700	3000	3500	65872	75879	65231	75132	64753	74603
7	6	6	6	6	6	50	8000	3200	3700	68932	78610	68120	77983	67645	77319
8	7	3	3	3	3	50	7500	2900	3600	66329	75439	65874	74973	65184	74306
9	7	4	4	4	4	50	7700	3100	3700	69372	78519	68726	78078	68104	77453
10	7	5	5	4	4	50	7900	3400	3800	71256	80325	70783	79790	70073	79135
11	7	5	5	5	5	70	8400	3800	4300	76320	89178	75760	88523	74897	87806
12	7	6	6	6	5	70	9000	4300	4850	83456	97206	82679	96645	81860	95890
13	7	7	7	7	7	70	9400	4700	5250	88732	103573	88011	102861	87325	102181
14	8	2	3	5	4	70	7500	3200	4100	76084	87230	75380	86523	74720	85972
15	8	5	3	3	3	70	8300	3900	4650	84620	96289	84056	95789	83467	95010
16	8	8	3	6	5	70	9300	4400	5200	95412	111981	94689	110238	93823	109378
17	8	8	8	8	5	70	10300	4800	5800	108092	124672	107459	123098	106249	122143
18	10	5	5	10	5	70	12800	5600	6700	128930	149806	127800	148720	126870	147689
19	10	8	7	10	5	70	13500	6200	7100	143987	187940	142138	186287	140356	184787
20	10	9	9	10	5	70	13800	6500	7400	152769	194250	151670	192893	150054	191200

Table 6.3: Some general data for large-size problems and the fitness values of the algorithms with Euclidean (EUD) and square Euclidean (SEUD) distances

Prob. No.	I	J	K	T	M	ub	B	f_k	S_k	SA		PSO		MFOA	
										EUD	SEUD	EUD	SEUD	EUD	SEUD
1	10	9	9	9	5	50	15000	7300	7900	172300	218943	170871	217219	169104	215672
2	10	10	10	10	5	50	16500	8100	8600	183025	238932	182092	237134	180893	235793
3	15	5	4	4	5	50	14000	7500	8400	187340	268334	186132	267298	185056	266109
4	15	5	4	8	5	50	15000	7600	8600	195629	288378	194210	287117	192879	285459
5	15	8	5	8	5	50	17500	7900	9200	213298	329568	211753	327985	209658	325195
6	15	10	5	10	5	50	19500	8200	9500	236643	379602	234786	376894	232100	374289
7	15	12	8	10	5	50	23000	9200	11000	267940	456732	264568	453487	262520	450132
8	15	15	10	10	5	50	26500	10700	13200	305632	523981	298317	519873	296745	516890
9	15	15	10	15	5	50	28000	12000	14500	332180	567532	329812	564786	327240	562098
10	30	5	5	5	5	50	26500	11600	13500	327935	557829	325231	555637	323467	553542
11	30	8	5	6	5	70	29000	13000	15500	343678	589760	341265	587943	339834	585378
12	30	10	5	8	5	70	32000	15000	17500	367823	632345	364790	629845	362145	626790
13	30	15	5	10	5	70	36000	16500	19500	442135	813945	439421	808457	436712	805783
14	30	20	5	10	5	70	45000	20000	25000	463214	867234	459811	864589	457121	861789
15	30	20	10	10	5	70	49500	23000	29000	497812	967321	494679	962478	491893	958620
16	40	10	10	10	6	70	53500	26500	33500	543760	1102340	538760	1098654	534271	1095387
17	40	20	10	20	6	70	60000	33000	39000	597290	1432500	593670	1428756	590980	1425721
18	40	30	10	20	6	70	65000	38000	44000	658930	1753205	654237	1749890	650678	1700980
19	40	40	20	40	6	70	74000	45000	51000	820987	2234670	814567	2187643	810983	2150300
20	50	30	20	50	6	70	86000	53000	59000	1078900	3157904	1072598	3152340	1068934	3148930

Table 6.4: The parameter levels of the three algorithms for *Prob. No. 11* of medium-size with Euclidean distance

	Level	μ_1	μ_2	NS	NP	NG
MFOA	1	1.5	1.5	3	30	200
	2	2	2	5	40	500
	3	2.5	2.5	7	50	1000
	Level	C_1	C_2	NP	NG	-
PSO	1	1.5	1.5	50	300	-
	2	2	2	70	500	-
	3	2.5	2.5	100	1000	-
	Level	α	$Inlo$	NG	-	-
SA	1	0.85	10	50	-	-
	2	0.9	20	70	-	-
	3	0.95	30	100	-	-

Table 6.5: The parameters levels of the three algorithms for *Prob. No. 11* of medium-size with square Euclidean distance

	Level	μ_1	μ_2	NS	NP	NG
MFOA	1	1.5	1.5	5	30	300
	2	2	2	7	40	500
	3	2.5	2.5	10	50	1000
	Level	C_1	C_2	NP	NG	-
PSO	1	1.5	1.5	60	400	-
	2	2	2	70	500	-
	3	2.5	2.5	100	1000	-
	Level	$Alpha$	$Inlo$	NG	-	-
SA	1	0.8	20	50	-	-
	2	0.9	30	70	-	-
	3	0.95	40	100	-	-

Table 6.6: The optimal levels of the parameters of the three algorithms for *Prob.*

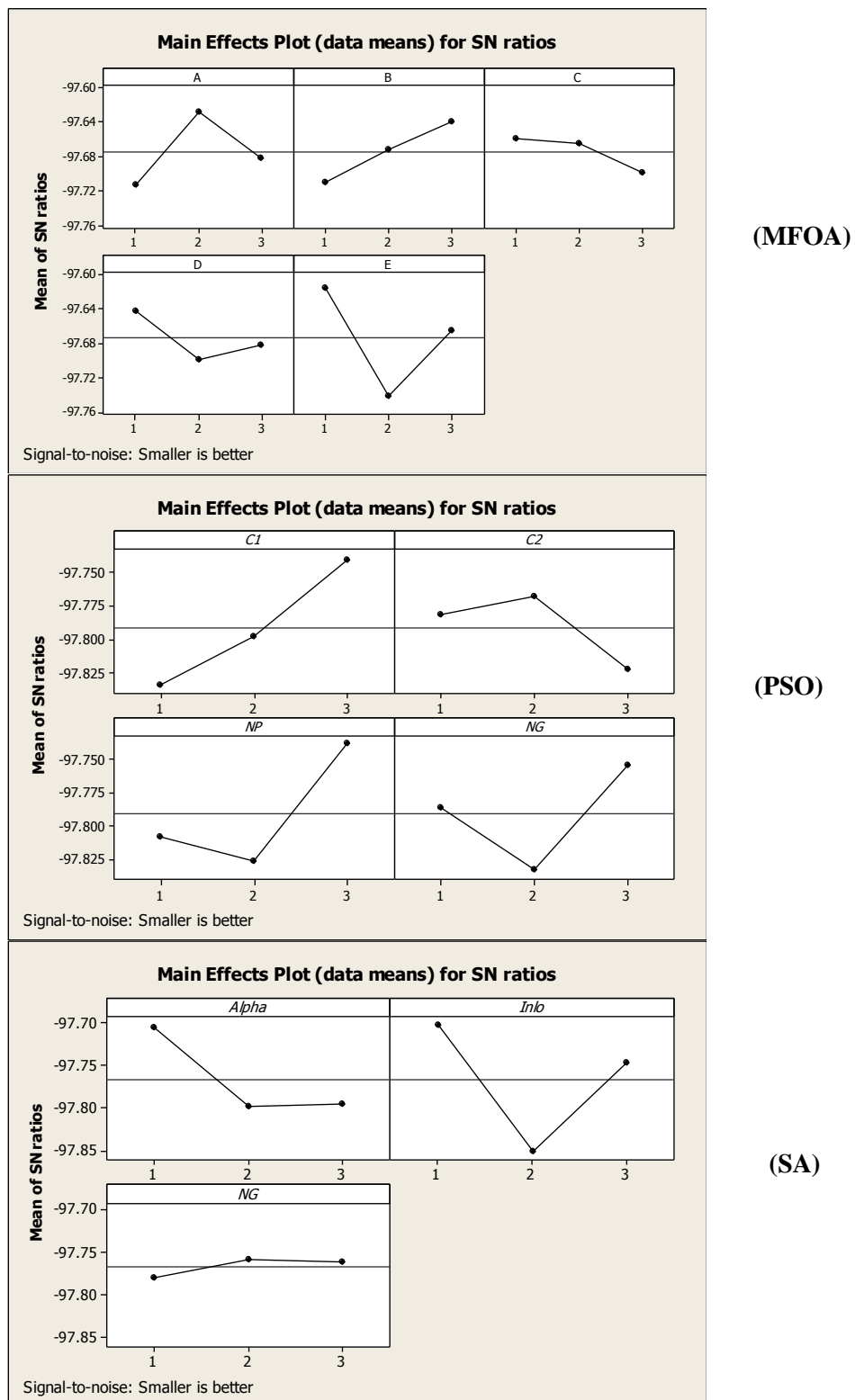
No. 11 of medium-size with Euclidean distance

Algorithms	Factors	Optimal levels
MFOA	μ_1	1.5
	μ_2	1.5
	<i>NS</i>	7
	<i>NP</i>	40
	<i>NG</i>	500
PSO	C_1	1.5
	C_2	2.5
	<i>NP</i>	70
	<i>NG</i>	500
SA	<i>Alpha</i>	0.9
	<i>Inlo</i>	30
	<i>NG</i>	50

Table 6.7: The optimal levels of the parameters of the three algorithms for *Prob.*

No. 11 of medium-size with square Euclidean distance

Algorithms	Factors	Optimal levels
MFOA	μ_1	2
	μ_2	2
	<i>NS</i>	10
	<i>NP</i>	40
	<i>NG</i>	300
PSO	C_1	2.5
	C_2	2.5
	<i>NP</i>	100
	<i>NG</i>	1000
SA	<i>Alpha</i>	0.9
	<i>Inlo</i>	40
	<i>NG</i>	70



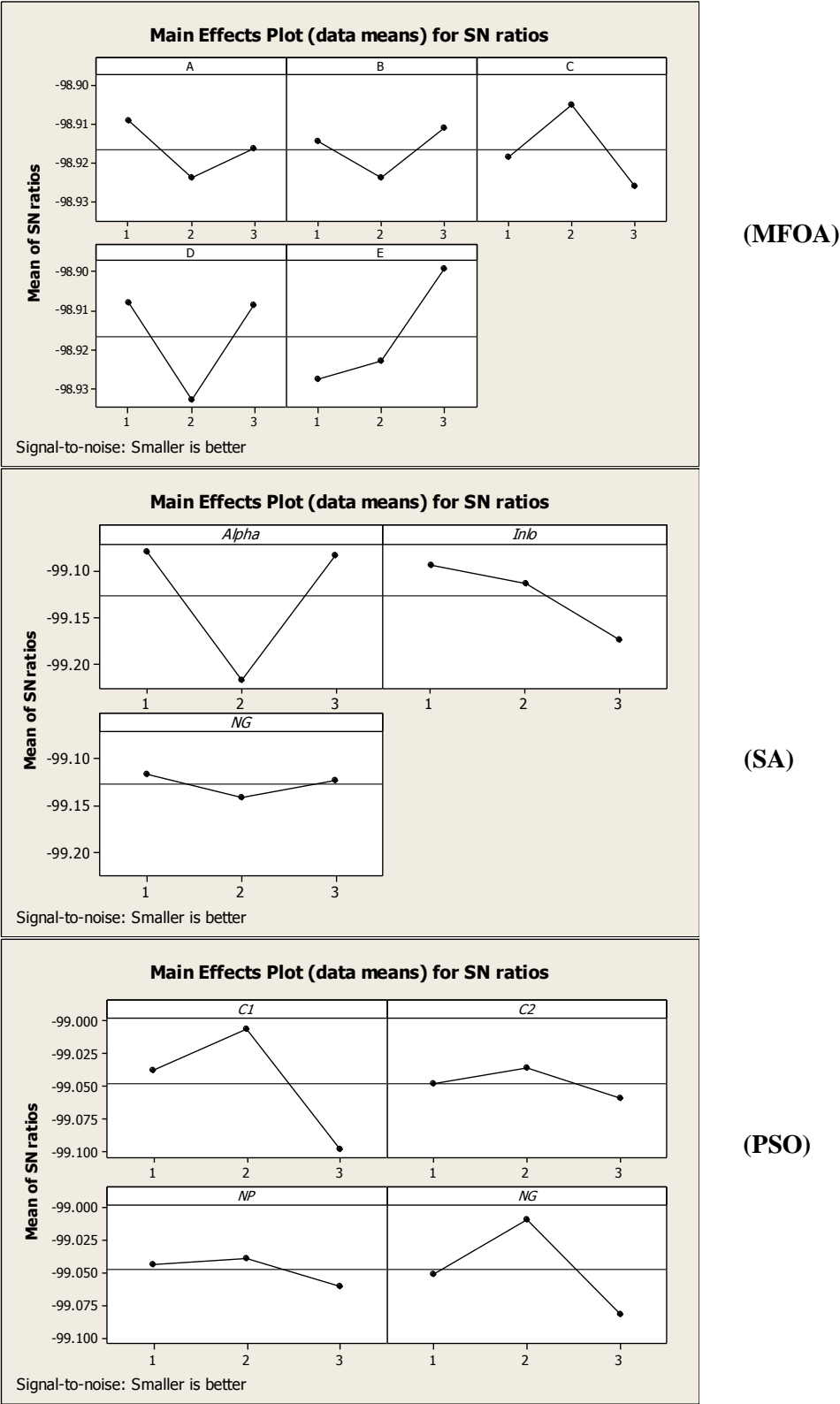


Figure 6.6: The mean S/N ratio plot for different levels of the parameters for *Prob.*

No. 11 of medium-size for MFOA, PSO, and SA with square Euclidean distance

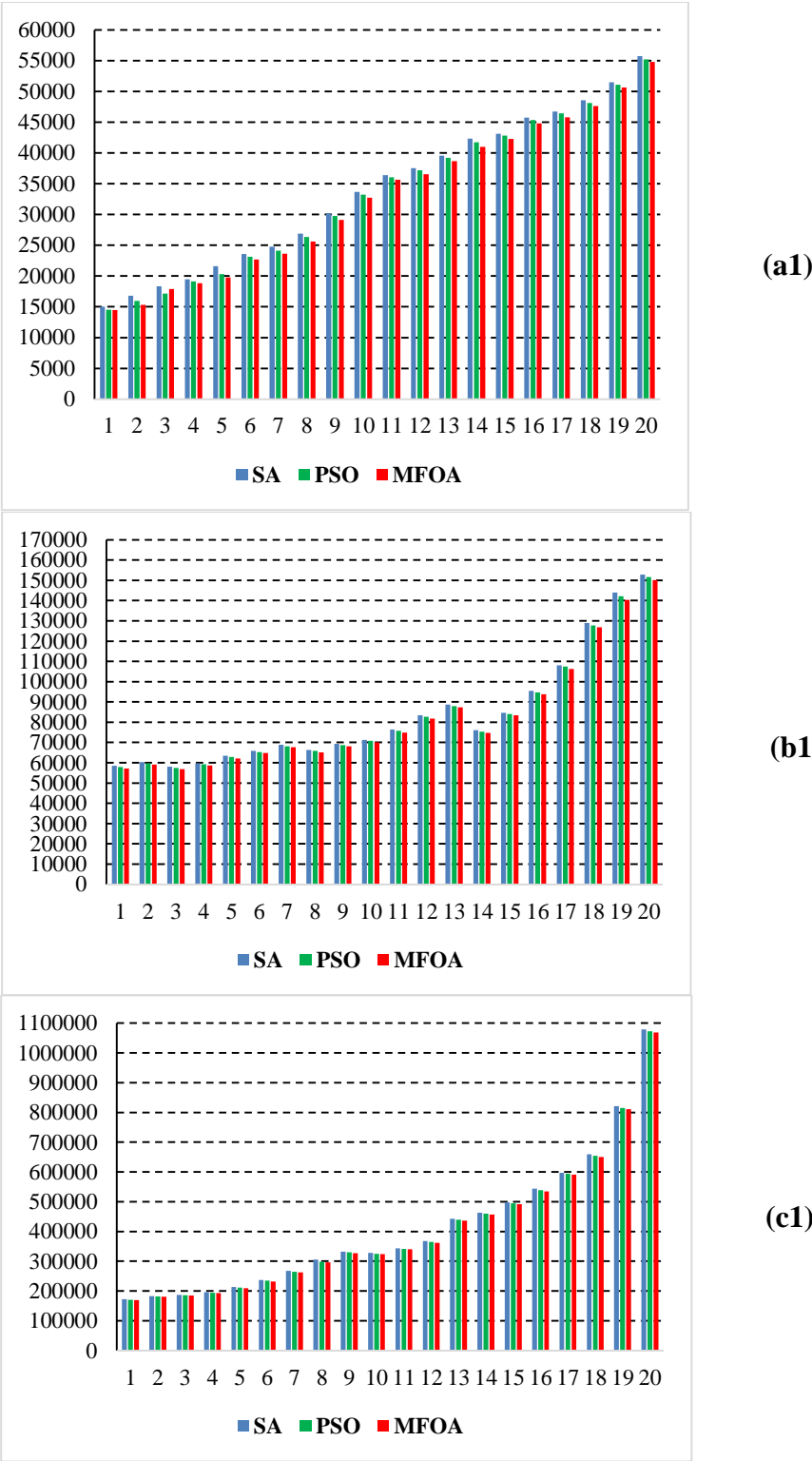


Figure 6.7: The graphical illustration of the fitness values of the algorithms with Euclidean distance for (a1) small-size, (b1) medium-size, and (c1) large-size problems

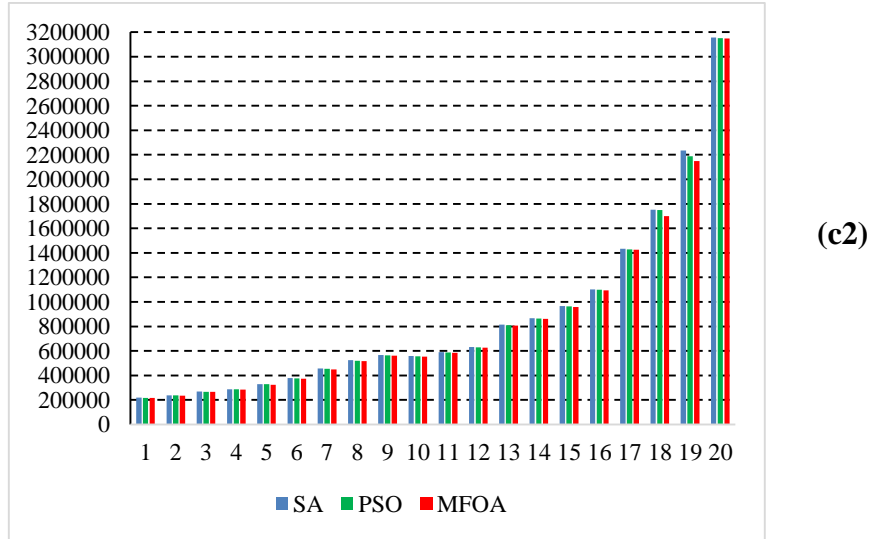
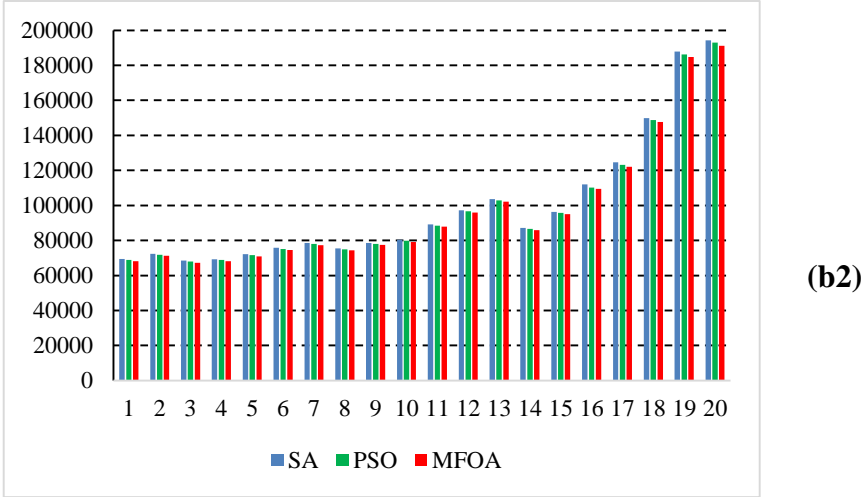
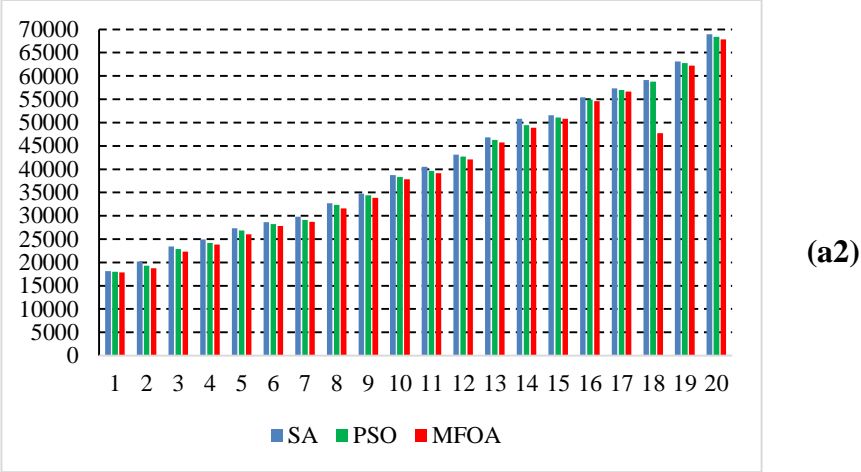


Figure 6.8: The graphical illustration of the fitness values of the algorithms with square Euclidean distance for (a2) small-size, (b2) medium-size, and (c2) large-size problems

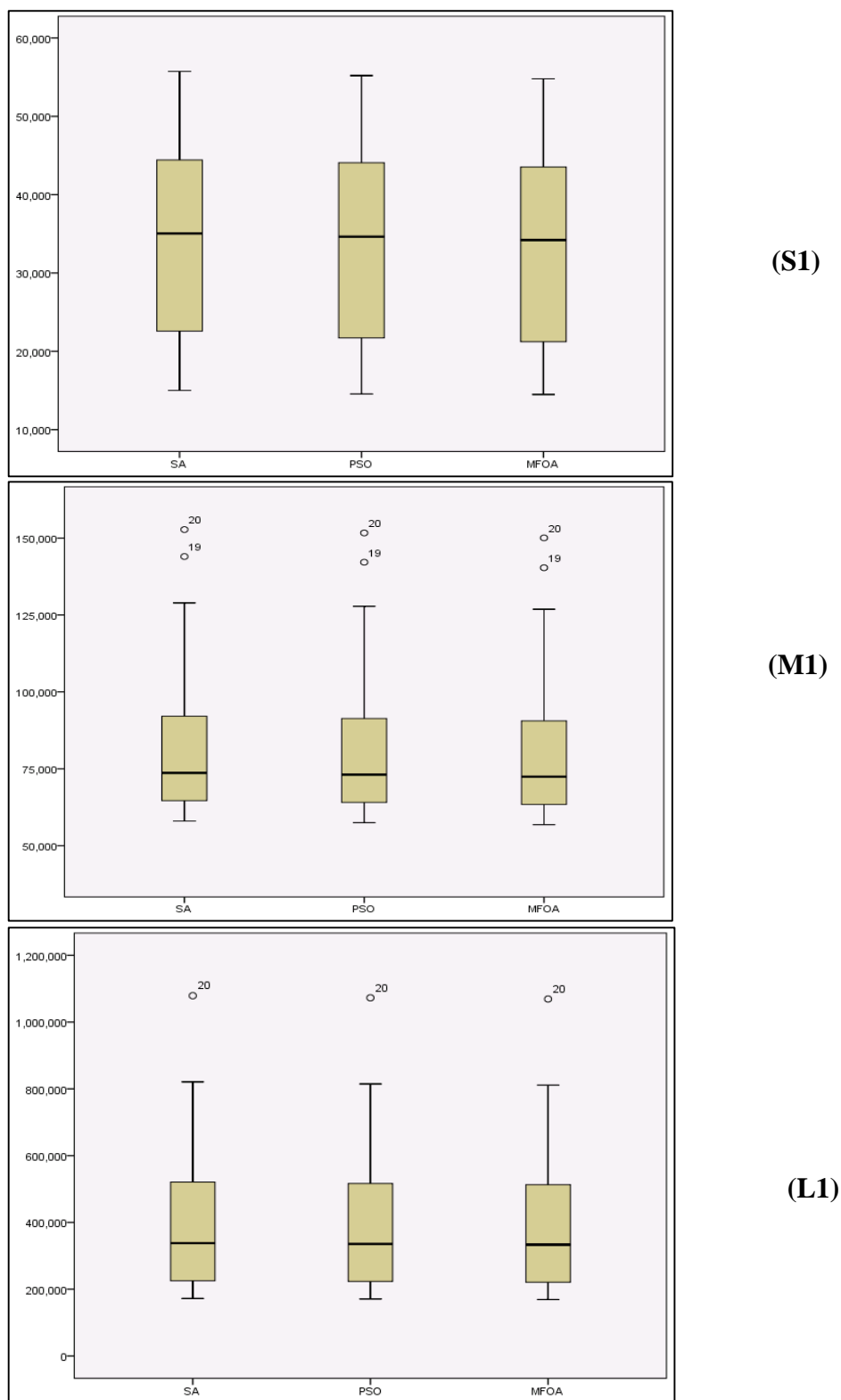


Figure 6.9: The box-plot of the proposed algorithms for (S1) small-size, (M1) medium-size, and (L1) large-size problems with Euclidean distances

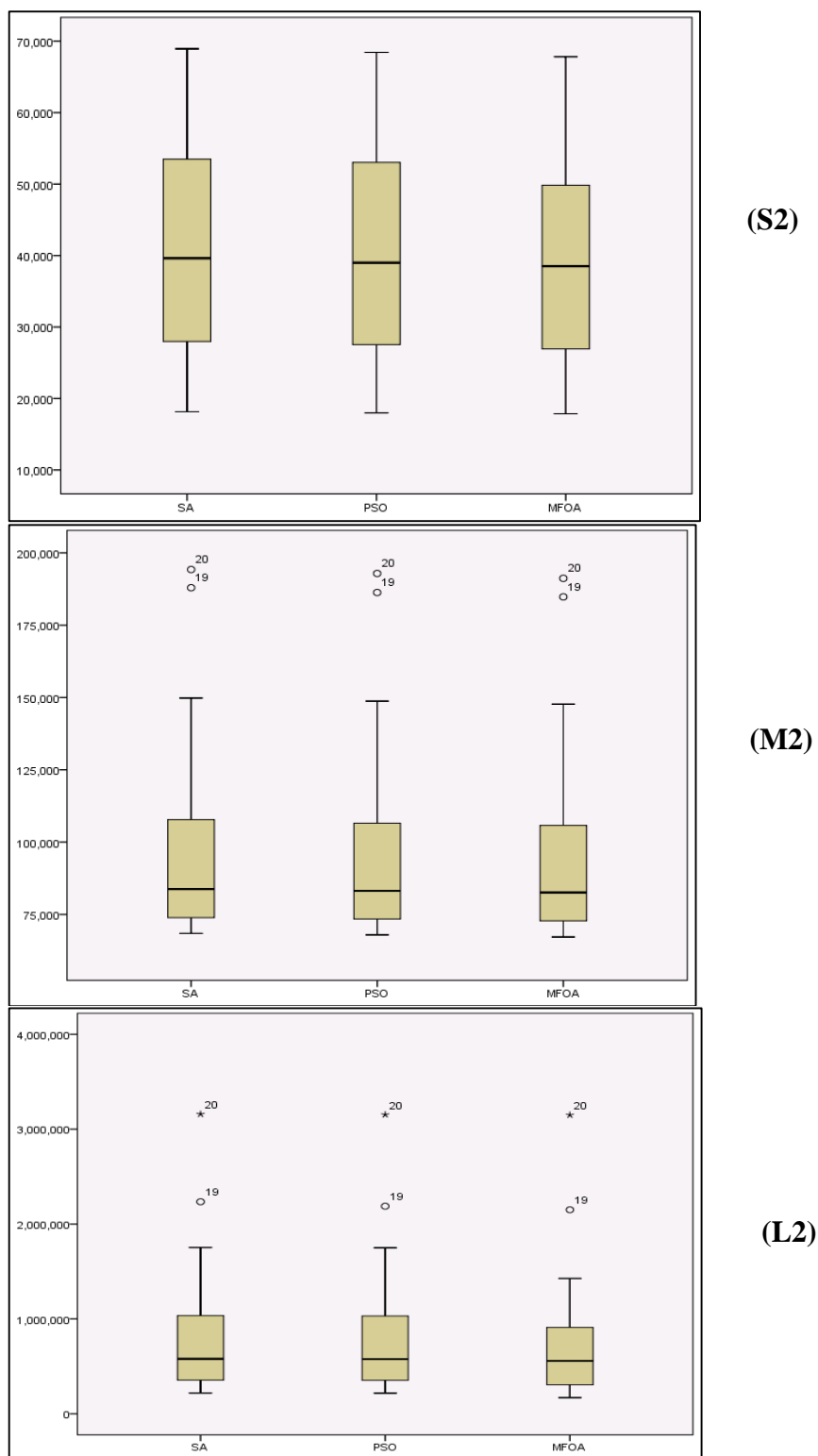


Figure 6.10: The box-plot of the proposed algorithms for (S2) small-size, (M2) medium-size, and (L2) large-size problems with square Euclidean

Table 6.8: The one-way ANOVA to compare the algorithms for small-size problems with Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	9495163	4747582	0.03	0.971
Error	57	9124042403	160070919	-	-
Total	59	9133537566	-	-	-

Table 6.9: The one-way ANOVA to compare the algorithms for small-size problems with square Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	24799330	12399665	0.05	0.947
Error	57	12985626072	227818001	-	-
Total	59	13010425402	-	-	-

Table 6.10: The one-way ANOVA to compare the algorithms for medium-size problems with Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	23675893	11837947	0.01	0.985
Error	57	45161958934	792315069	-	-
Total	59	45185634827	-	-	-

Table 6.11: The one-way ANOVA to compare the algorithms for medium-size problems with square Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	25373700	12686850	0.01	0.991
Error	57	79790907182	1399840477	-	-
Total	59	79816280883	-	-	-

Table 6.12: The one-way ANOVA to compare the algorithms for large-size problems with Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	322846471	161423235	0	0.997
Error	57	3.16E+12	55521298438	-	-
Total	59	3.17E+12	-	-	-

Table 6.13 The one-way ANOVA to compare the algorithms for large-size problems with square Euclidean distance

Source	DF	SS	MS	F	P-value
Factor	2	98370821830	49185410915	0.09	0.916
Error	57	3.20E+13	5.61E+11	-	-
Total	59	3.21E+13	-	-	-

Table 6.14: CPU times (s) taken by each algorithm to solve small-size problems

<i>Prob. No.</i>	SA		PSO		MFOA	
	EUD	SEUD	EUD	SEUD	EUD	SEUD
1	12.82	12.43	9.86	9.63	8.89	8.74
2	13.23	13.12	10.11	9.94	9.67	9.42
3	15.4	15.08	13.33	13.19	11.83	11.78
4	17.32	16.88	14.49	14.38	12.59	12.29
5	18.13	17.91	16.87	16.59	15.69	15.38
6	14.34	14.06	12.37	11.93	10.53	10.48
7	16.83	16.66	15.21	14.88	13.72	13.4
8	18.08	17.73	16.29	16.12	15.38	15.26
9	22.88	22.59	19.63	19.27	18.33	18.05
10	21.75	21.38	19.42	19.28	17.63	17.34
11	23.6	23.44	20.89	20.67	18.63	18.45
12	25.81	25.39	21.77	21.63	20.12	19.84
13	24.79	24.43	21.2	20.96	19.83	19.62
14	26.72	26.54	22.39	22.18	21.4	21.25
15	25.61	25.42	22.11	21.82	20.92	20.63
16	29.33	29.12	24.81	24.51	23.61	23.38
17	34.72	33.89	26.22	25.77	24.32	24.19
18	31.72	31.45	25.19	24.88	23.96	23.69
19	33.27	33.12	25.89	25.72	24.21	23.92
20	41.13	40.37	29.39	28.68	28.31	28.03

Table 6.15: CPU times (s) taken by each algorithm to solve medium-size problems

<i>Prob. No.</i>	SA		PSO		MFOA	
	EUD	SEUD	EUD	SEUD	EUD	SEUD
1	56.81	56.2	48.71	47.89	44.54	44.21
2	59.63	59.27	49.73	49.32	45.39	45.19
3	49.73	49.44	46.53	46.11	43.21	42.78
4	50.46	49.71	47.23	47.02	43.88	43.56
5	57.96	57.37	48.34	48.19	44.61	44.25
6	61.39	60.87	50.43	50.16	47.76	47.41
7	65.19	64.83	53.95	53.72	51.11	50.76
8	59.43	59.21	50.32	50.01	47.61	47.32
9	63.11	62.81	52.44	52.19	50.69	50.45
10	66.29	65.88	55.29	54.79	51.13	50.98
11	68.49	68.12	58.71	58.17	53.58	53.29
12	71.99	71.69	62.44	62.28	56.43	56.32
13	74.12	73.8	65.39	65.14	59.59	58.95
14	68.88	68.37	60.48	60.21	55.74	55.36
15	71.13	70.79	63.45	62.99	56.78	56.49
16	75.44	74.87	66.91	66.37	58.22	57.74
17	79.33	78.86	70.09	69.67	63.44	62.93
18	83.77	83.32	74.45	73.91	69.56	69.23
19	86.22	85.71	76.23	75.87	71.29	70.82
20	89.62	89.32	78.21	77.89	73.77	73.34

Table 6.16: CPU times (s) taken by each algorithm to solve large-size problems

<i>Prob. No.</i>	SA		PSO		MFOA	
	EUD	SEUD	EUD	SEUD	EUD	SEUD
1	93.22	92.67	80.72	80.21	76.26	75.88
2	98.56	98.06	83.55	83.18	78.93	78.41
3	86.26	85.78	73.45	73.11	65.39	64.94
4	87.77	87.33	74.32	73.89	66.23	65.89
5	90.33	90.04	76.55	76.18	69.49	69.05
6	94.22	93.85	78.37	77.9	71.78	71.26
7	98.82	98.37	82.53	82.24	74.93	74.52
8	103.28	102.82	86.77	86.39	78.49	78.27
9	109.03	108.79	92.91	92.61	83.86	83.47
10	100.32	99.79	89.03	88.78	81.3	80.88
11	103.69	103.42	91.49	91.09	84.63	84.18
12	108.72	108.25	94.55	94.31	86.77	86.39
13	113.19	112.83	98.48	97.82	89.82	89.51
14	122.88	121.09	108.76	108.17	99.26	98.89

Table 6.16 Continued

15	131.49	130.74	117.34	116.93	108.39	107.91
16	135.39	134.88	121.64	121.27	111.25	110.83
17	152.65	152.22	137.32	136.56	125.75	124.86
18	174.92	174.04	159.83	158.91	141.77	140.82
19	212.33	210.67	188.49	187.21	172.09	171.37
20	215.28	214.51	191.98	191.15	176.81	175.85

6.4.2 A Case Study

An extension of the case study considered in the previous chapter is investigated in this section where the shortages were allowed. The distances among the retailers and the distributors were assumed to be Euclidean. Table 6.17 shows the backordering and lost sale costs of the case study explained in section 5.4.3 where the data sheet collected from the company is displayed in Appendix B. The rest of parameters values are given as the same proposed in Chapter 5. In case of shortage, 90 percent of the customers facing shortages stay in the system to receive their orders to be considered as backorder ($\beta = 0.9$) and the rest of customers ($1 - \beta = 0.1$) leave the system to receive their orders somewhere else to be considered as lost sale.

Table 6.17: The backordering and lost sale costs of the case study explained in section 5.4.3

Product	μ_{1jk1}	μ_{1jk2}	μ_{1jk3}	μ'_{1jk1}	μ'_{1jk2}	μ'_{1jk3}
<i>MDF (sheet)</i>	50000	60000	70000	90000	60000	800
<i>Adhesive (Kg)</i>	50000	60000	80000	90000	50000	800
<i>HGF (120m)</i>	30000	40000	60000	100000	90000	600
<i>PVC (120m)</i>	40000	20000	40000	40000	60000	400

The total supply chain cost, the optimal locations of the vendors and the optimal order quantities purchased by retailers from distributors obtained by MFOA are obtained as follows, where the elapsed CPU time is equal to 35.632 seconds.

$Tc_{MFOA} = 472,000,000$ IT includes:

Transportation cost (TrC) = 201,550,000 IT

Holding cost (HoC) = 96,180,000 IT

Shortage cost (ShC) = 0

Purchasing cost (Pc) = 174,270,000 IT

$y_1 = (30, 53)$; $y_2 = (36, 61)$

$Q_{1MDFkt} = [330, 400, 380, 330, 400, 490]$

$Q_{1HMAkt} = [500, 380, 410, 420, 520, 430]$

$Q_{1HGFkt} = [460, 450, 470, 450, 430, 420]$

$Q_{1PVCKt} = [450, 480, 500, 530, 450, 470]$

$Q_{2MDFkt} = [450, 390, 380, 410, 510, 480]$

$Q_{2HMAkt} = [490, 440, 430, 470, 500, 490]$

$Q_{2HGFkt} = [480, 470, 450, 420, 440, 480]$

$Q_{2PVCKt} = [520, 470, 450, 420, 460, 450]$

$Q_{3MDFkt} = [480, 420, 400, 380, 510, 450]$

$Q_{3HMAkt} = [470, 460, 420, 490, 440, 430]$

$Q_{3HGFkt} = [500, 490, 400, 440, 490, 450]$

$Q_{3PVCKt} = [400, 430, 490, 500, 510, 390]$

Figure 6.11 shows a representation of the optimal locations of the producers among the retailers obtained by MFOA for the proposed case study.

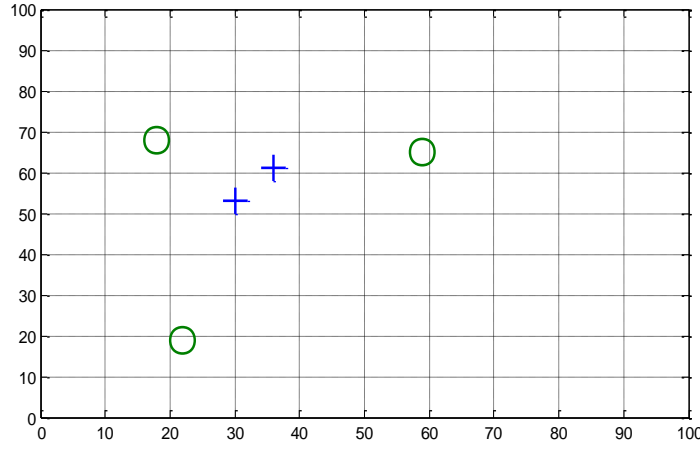


Figure 6.11: The optimal location of the producers among the retailers obtained by MFOA for the case study

The total supply chain cost, the optimal locations of the vendors and the optimal order quantities purchased by retailers from distributors obtained by PSO are obtained as follows, where the elapsed CPU time is equal to 46.127 seconds.

$Tc_{PSO} = 516,450,000$ IT includes:

Transportation cost (TrC) = 239,240,000 IT,

Holding cost (HoC) = 93,375,000 IT,

Shortage cost (TrC) = 6,535,000 IT consists of 5,175,000 IT backordering cost and 1,360,000 IT lost sale cost,

Purchasing cost (Pc) = 177,300,000 IT and

$$y_1 = (33, 29); y_2 = (51, 38)$$

$$Q_{1MDFkt} = [70, 580, 570, 100, 580, 570]$$

$$Q_{1HMAkt} = [90, 90, 550, 490, 510, 490]$$

$$Q_{1HGFkt} = [440, 530, 490, 30, 540, 560]$$

$$Q_{1PVCkt} = [80, 560, 530, 80, 490, 580]$$

$$Q_{2MDFkt} = [90, 60, 540, 540, 540, 520]$$

$$Q_{2HMAkt} = [530, 540, 590, 520, 570, 580]$$

$$Q_{2HGFkt} = [540, 560, 490, 580, 580, 400]$$

$$Q_{2PVCKt} = [30, 590, 540, 490, 530, 570]$$

$$Q_{3MDFkt} = [80, 530, 570, 530, 580, 590]$$

$$Q_{3HMAkt} = [550, 520, 550, 520, 580, 590]$$

$$Q_{3HGFkt} = [530, 590, 590, 470, 510, 480]$$

$$Q_{3PVCKt} = [40, 40, 480, 580, 560, 550]$$

Figure 6.12 shows a representation of the optimal locations of the producers among the retailers obtained by PSO for the proposed case study.

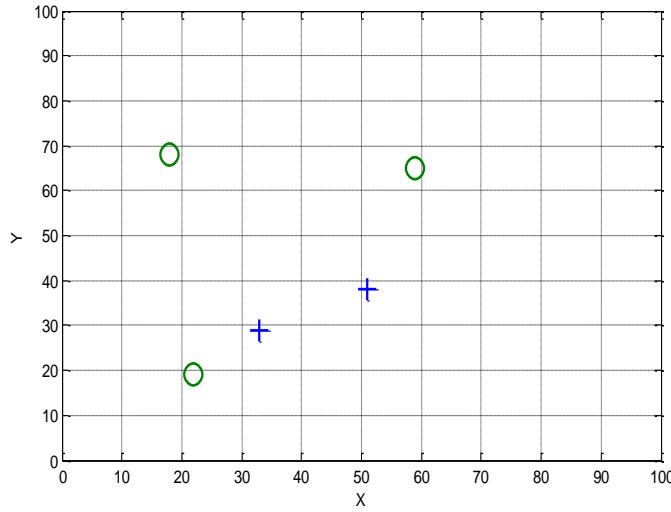


Figure 6.12: The optimal location of the producers among the retailers obtained by PSO for the case study

The total supply chain cost, the optimal locations of the vendors and the optimal order quantities purchased by retailers from distributors obtained by SA are obtained as follows, where the elapsed CPU time is equal to 73.503 seconds.

$$Tc_{SA} = 588,754,000 \text{ IT includes:}$$

$$\text{Transportation cost (} TrC \text{)} = 289,940,000 \text{ IT,}$$

$$\text{Holding cost (} HoC \text{)} = 113,594,000 \text{ IT,}$$

Shortage cost (TrC) = 5,240,000 IT consists of 4,185,000 IT backordering cost and 1,055,000 IT lost sale cost,

Purchasing cost (Pc) = 179,980,000 IT and

$$y_1 = (33, 29); y_2 = (51, 38)$$

$$Q_{1MDFkt} = [60, 590, 570, 100, 580, 570]$$

$$Q_{1HMAkt} = [80, 70, 580, 530, 510, 490]$$

$$Q_{1HGFkt} = [440, 530, 490, 40, 540, 580]$$

$$Q_{1PVCKt} = [80, 590, 530, 80, 540, 580]$$

$$Q_{2MDFkt} = [90, 70, 540, 560, 520, 520]$$

$$Q_{2HMAkt} = [530, 540, 590, 520, 570, 580]$$

$$Q_{2HGFkt} = [540, 560, 500, 580, 580, 400]$$

$$Q_{2PVCKt} = [50, 590, 540, 490, 530, 570]$$

$$Q_{3MDFkt} = [80, 530, 570, 530, 580, 590]$$

$$Q_{3HMAkt} = [550, 520, 550, 520, 580, 590]$$

$$Q_{3HGFkt} = [530, 590, 590, 470, 510, 480]$$

$$Q_{3PVCKt} = [40, 280, 490, 580, 580, 560]$$

Figure 6.13 shows a representation of the optimal locations of the producers among the retailers obtained by SA for the proposed case study.

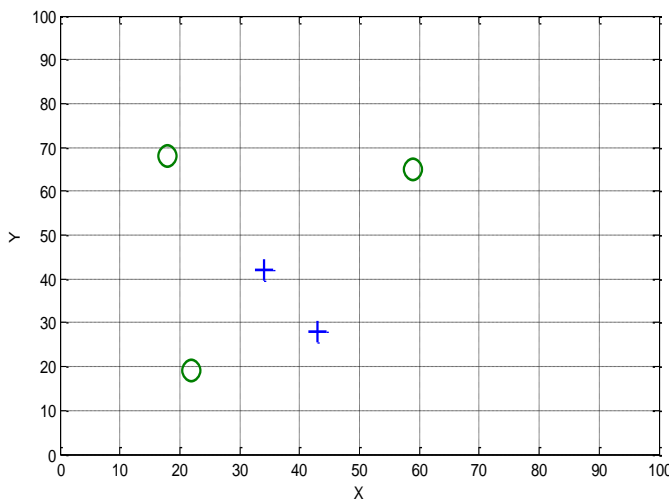


Figure 6.13: The optimal location of the producers among the retailers obtained by SA for the case study

From the results obtained by three algorithms on the case study, FOA has shown to be better algorithm than PSO and SA in terms of the total supply chain cost and CPU time.

6.5 Summary

In this chapter, a mixed binary-integer programming model was developed for a two-echelon distributor-retailer supply chain network problem with multiple products producing in multiple periods (seasons) and shortages as a combination of backorders and lost sales. The distributors produced the products and stored them in their warehouses to respond to the orders made by the retailers in different periods, where the products were sold under two all-unit and incremental quantity discount policies. The distributors' warehouse spaces, the total budget for purchasing the products, and also the production capacity of the distributors were limited. The aim was to find the optimal order quantities and the optimal location coordinates of the distributors among the retailers so as the total cost of the supply chain network including transportation, holding, shortage, and purchases was minimized. To solve the problem, a MFOA was utilized. As there was no benchmark available in the literature, two other meta-heuristics, namely PSO and SA was employed to validate the results obtained. Some problem instances of three sizes were randomly generated in addition to a case study were to evaluate the efficiency of the algorithms, after parameter calibration of the parameters of all algorithms using Taguchi method. Statistical comparisons of the means of the fitness functions in all problem instances in which Euclidean and square Euclidean distances were used showed that there were no differences. In other words, the results obtained using MFOA were validated by the other two algorithms.

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusion

In this work, first a mixed-integer binary mathematical programming model was developed for a multi-objective multi-item multi-period inventory control problem with AUD and IQD discount policies. To make the model more realistic, budget and order quantity were constrained where the shortages were allowed and contained backorder and lost sale. In the proposed mathematical model, a binary variable was used to model the order quantity of an item in a period. The model was close to the inventory control models in which order and sale processes were carried out in the same season. Due to adopting decisions related to a certain department of production planning (extending warehouse or building a new manufacturing line), the manager decided to build a new warehouse for the ordered items. We followed to minimize the total inventory system cost over a finite horizon in addition to minimizing the total storage space to determine the optimal order quantities of the products for which a fuzzy weighted combination was defined as the objective function. The weights of the objectives were considered as triangular fuzzy numbers. Two meta-heuristics called MOPSO and MOGA were utilized to solve the proposed NP-hard problem. Furthermore, not only MATLAB software was applied to code the proposed meta-heuristic algorithms, but also Taguchi method was implemented to calibrate the parameters of the meta-heuristics. The results showed that for the 10 specific problems and the case study the MOPSO performed better than the MOGA in terms of the fitness function values.

Next, the model was developed to formulate a mixed-integer binary two-echelon inventory-supply chain system for a facility location allocation problem in which each vendor had his/her own warehouse. The products were produced by the

vendors and then sent to their warehouses to meet a demand. The buyers purchased the products under AUD discount strategy proposed by the vendors where the total available budget was restricted. The model was formulated under the constraints of the total warehouse capacity and the production limitation of each vendor. The distance among the buyers and vendors were supposed to be Euclidean. The aim was to obtain the optimal order quantities purchased by the buyers from vendors in addition to determining the optimal locations of vendors such that the total inventory-supply chain costs including transportation cost, holding cost and purchasing cost were minimized. To solve the proposed supply chain model, a MPSO algorithm was employed where a GA was utilized to validate the results of the proposed algorithm. Taguchi method was also applied to set the parameters of the two algorithms. The results of the algorithms showed the MPSO has a better performance than the GA in terms of the objective function on the generated instances of the three categories.

Third, a multi-product multi-period inventory-supply chain was derived for a location allocation problem to be considered as a mixed-integer binary nonlinear mathematical programming model. The total budget, the truck capacity and the warehouse and production capacities of each distributor were restricted. The distributors sold their products under both AUD and IQD discount policies. Also, the distances among the distributors and retailers were considered to be Euclidean and Square Euclidean. Shortages were allowed and included backorder and lost sale. The objective was to determine the optimal coordinates of the distributors among the retailers and also the optimal order quantities purchased by retailers from the distributors so that the total inventory-supply chain costs became minimized. To solve the problem, a modified fruit fly optimization algorithm (MFOA) was utilized. As there was no benchmark available in the literature, two other meta-heuristics,

namely PSO and SA were employed to validate the results obtained. Some problem instances of three sizes were randomly generated to evaluate the efficiency of the algorithms, after parameter calibration of the parameters of all algorithms using Taguchi method. Statistical comparisons of the means of the fitness functions in all problem instances in which Euclidean and square Euclidean distances were used showed that there were no differences. Additionally, the results of the algorithms on a case study showed that the MFOA had better performance than the other algorithms.

The achievements of the objectives proposed in Chapter 1 were addressed accordingly in this thesis where the first and second objectives were addressed in Chapter 4 and 5 respectively. Furthermore, the third objective was addressed by the proposed algorithms explained in Chapters 4, 5 and 6 as well.

7.2 Contributions

The contributions of the study are discussed into three parts as follows:

The first contribution of the problem is considering a new bi-objective multi-item multi periodic inventory control model where some items were purchased under AUD and the other items are bought from IQD. The demands vary in different periods, the budget is limited, the orders were placed in batch sizes, and shortages in combination of backorder and lost sale are considered. The goal was to find the optimum order quantities of the items in each period such that the total inventory cost and the total required warehouse space were minimized simultaneously. Since it is not easy for the managers to allocate the crisp values to the weights of the objectives in a decision making process, considering these weights as fuzzy numbers will be taken as an advantage.

The second novel contribution of the study is that the integrated supply chain expressed in this work simultaneously determines two types of decision variables: (i) the locations of the vendors in a certain area among the buyers with fixed locations and (ii) the allocation: the order quantities of the products at each period made by the buyers from the vendors. The total available budget for purchasing the products and also the total vendors' warehouse space are constrained. Moreover, the distance between the buyers and the vendors was assumed as Euclidean distance.

The third contribution of this research study was briefly expressed as follows. Firstly, a novel mixed-integer binary nonlinear model of a two-echelon supply chain network for multi-product location allocation-inventory control problem in multiple periods was provided where several constraints are considered to make the model applicable to a closer to reality problem. Also, the shortages are allowed where a fraction of products were satisfied as backordering and a fraction was lost sale. Secondly, for the first time in the literature a modified version of the Fruit Fly optimization algorithm called MFOA was developed to solve inventory-supply chain and location-allocation problem. Lastly, a design of experiment method, i.e. the Taguchi approach, was applied to tune the parameters of the MFOA.

7.3 Implementations to the study

The applications of the proposed models in the real world can be described as follows:

- I. Fashion companies those producing the clothes for satisfying the demands for some particular periods. In this case, the items will arrive to the market in a specific period and will be out of stock in another period.

- II. Heating and cooling companies those manufacture products for some particular seasons. Air-conditioners are produced for the hot seasons and heating equipment to be used for the cool seasons.
- III. Those companies have two-echelon supply chain network and schedule their manufacturing process for several products for a specific time-period. The case study performed in this study is a good example for these sorts of the companies.
- IV. Those companies aim to locate a number of branches in some potential locations where the potential demands are identified already.

7.4 Recommendations for Future Research

Some recommendations for future works are to expand the model to cover some terms as follows:

- i. A multi-echelon supply chain environment of the inventory problem can be considered.
- ii. Both of the proposed inventory control problem and inventory-supply chain problem can be investigated for the models with fuzzy or stochastic demands.
- iii. The inflation and the time value of the money rates will be another recommendation for future works that can be taken into account.
- iv. To find the locations of the vendors among the buyers, the inventory-supply chain location allocation problems can be modeled with different kind of distance functions such as Rectilinear distances, Manhattan and d-norm instead of using Euclidean and Square Euclidean.
- v. In order to simplify the proposed inventory and supply chain problems, a zero lead for replenishment/production time was assumed. As a

recommendation for the future, lead time can be considered for inventory-supply chain location allocation problem in both stochastic and deterministic environments.

- vi. Redundancy allocation problem (RAP) is a common problem which has recently been taken into account by many researchers. A mixed inventory RAP problem or inventory-supply chain RAP model can be attended as a recommendation for the future study.

7.5 Shortcomings and Strengths

The shortcomings of the thesis can be as:

- Finding a real life company in Malaysia to collect data
- Increasing the number of dimensions (variables) of the problem causes a increase in running time of algorithms

The main strength of the study is:

The models are applicable for many companies in the real world.

REFERENCES

- Aarts, E., & Korst, J. (1988). *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*, John Wiley & Sons.
- Abad, P. (1988). Joint price and lot-size determination when supplier offers incremental quantity discounts. *Journal of the Operational Research Society*, 39(6), 603-607.
- Abad, P. L. (1988). Determining Optimal Selling Price and Lot Size When the Supplier Offers All- Unit Quantity Discounts. *Decision Sciences*, 19(3), 622-634.
- Abdel-Malek, L., Montanari, R., & Morales, L. C. (2004). Exact, approximate, and generic iterative models for the multi-product newsboy problem with budget constraint. *International Journal of Production Economics*, 91(2), 189-198.
- Abiri, M., & Yousefli, A. (2011). An application of probabilistic programming to the fuzzy location-allocation problems. *International Journal of Advanced Manufacturing Technology*, 53(9), 1239-1245.
- Abolhasani, Z. H., Marian, R. M., & Loung, L. (2013). Optimization of multi-commodities consumer supply chains part II: Simulation modeling. *Journal of Computer Science*, 9(12), 1830.
- Absi, N., Detienne, B., & Dauzère-Pérès, S. (2013). Heuristics for the multi-item capacitated lot-sizing problem with lost sales. *Computers & Operations Research*, 40(1), 264-272.
- Absi, N., & Kedad-Sidhoum, S. (2009). The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs. *Computers & Operations Research*, 36(11), 2926-2936.
- Ahmadi-Javid, A., & Seddighi, A. H. (2013). A location-routing problem with disruption risk. *Transportation Research Part E: Logistics and Transportation Review*, 53, 63-82.
- Ahmadzadeh, E., & Vahdani, B. (2017). A location-inventory-pricing model in a closed loop supply chain network with correlated demands and shortages under a periodic review system. *Computers & Chemical Engineering*, 101, 148-166.
- Ahmed, S., Çakmak, U., & Shapiro, A. (2007). Coherent risk measures in inventory problems. *European Journal of Operational Research*, 182(1), 226-238.
- Akbari Kaasgari, M., Imani, D. M., & Mahmoodjanloo, M. (2017). Optimizing a vendor managed inventory (VMI) supply chain for perishable products by considering discount: Two calibrated meta-heuristic algorithms. *Computers & Industrial Engineering*, 103, 227-241.
- Alikar, N., Mousavi, S. M., Ghazilla, R. A. R., Tavana, M., & Olugu, E. U. (2017). A bi-objective multi-period series-parallel inventory-redundancy allocation problem with time value of money and inflation considerations. *Computers & Industrial Engineering*, 104, 51-67.

- Aloulou, M. A., Dolgui, A., & Kovalyov, M. Y. (2014). A bibliography of non-deterministic lot-sizing models. *International Journal of Production Research*, 52(8), 2293-2310.
- Arabzad, S. M., Ghorbani, M., & Hashemkhani Zolfani, S. (2015). A multi-objective robust optimization model for a facility location-allocation problem in a supply chain under uncertainty. *Engineering Economics*, 26(3), 227-238.
- Azadeh, A., Elahi, S., Farahani, M. H., & Nasirian, B. (2017). A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. *Computers & Industrial Engineering*, 104, 124-133.
- Balasubbareddy, M. (2016). Multi-objective optimization in the presence of ramp-rate limits using non-dominated sorting hybrid fruit fly algorithm. *Ain Shams Engineering Journal*, 7(2), 895-905.
- Bandyopadhyay, S., & Bhattacharya, R. (2013). Solving a tri-objective supply chain problem with modified NSGA-II algorithm. *Journal of Manufacturing Systems*, 33(1), 41-50.
- Basnet, C., & Leung, J. M. Y. (2005). Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32(1), 1-14.
- Baykasoglu, A., & Gocken, T. (2011). Solving fully fuzzy mathematical programming model of EOQ problem with a direct approach based on fuzzy ranking and PSO. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 22(5, 6), 237-251.
- Bazan, E., Jaber, M. Y., & Zanoni, S. (2016). A review of mathematical inventory models for reverse logistics and the future of its modeling: An environmental perspective. *Applied Mathematical Modelling*, 40(5), 4151-4178.
- Behret, H., & Kahraman, C. (2012). Single-Period Inventory Models with Discrete Demand Under Fuzzy Environment. *Multiple-Valued Logic and Soft Computing*, 18(3-4), 371-386.
- Ben-Daya, M., & Raouf, A. (1993). On the constrained multi-item single-period inventory problem. *International Journal of Operations & Production Management*, 13(11), 104-112.
- Benson, H. (2000). Nonlinear Multiobjective Optimization (Kaisa M. Miettinen). *SIAM REVIEW*, 42(2), 339-340.
- Benton, W. (1985). Multiple price breaks and alternative purchase lot-sizing procedures in material requirements planning systems. *International Journal of Production Research*, 23(5), 1025-1047.
- Bhunia, A. K., Shaikh, A. A., & Cárdenas-Barrón, L. E. (2017). A partially integrated production-inventory model with interval valued inventory costs, variable demand and flexible reliability. *Applied Soft Computing*, 55, 491-502.

- Boonmee, A., & Sethanan, K. (2016). A GLNPSO for multi-level capacitated lot-sizing and scheduling problem in the poultry industry. *European Journal of Operational Research*, 250(2), 652-665.
- Bozorgi-Amiri, A., Jabalameli, M. S., Alinaghian, M., & Heydari, M. (2012). A modified particle swarm optimization for disaster relief logistics under uncertain environment. *The International Journal of Advanced Manufacturing Technology*, 60(1-4), 357-371.
- Brahimi, N., Absi, N., Dauzère-Pérès, S., & Nordli, A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*. doi.org/10.1016/j.ejor.2017.05.008.
- Cárdenas-Barrón, L. E., González-Velarde, J. L., & Treviño-Garza, G. (2015). A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem. *Computers & Operations Research*, 64, 225-232.
- Cardona-Valdés, Y., Álvarez, A., & Pacheco, J. (2014). Metaheuristic procedure for a bi-objective supply chain design problem with uncertainty. *Transportation Research Part B: Methodological*, 60, 66-84.
- Carrizosa, E., Conde, E., Munoz-Marquez, M., & Puerto, J. (1995). The generalized Weber problem with expected distances. *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, 29(1), 35-57.
- Carrizosa, E., Munoz-Marquez, M., & Puerto, J. (1998). A note on the optimal positioning of service units. *Operations Research*, 46(1), 155-156.
- Carvalho, D. M., & Nascimento, M. C. V. (2016). Lagrangian heuristics for the capacitated multi-plant lot sizing problem with multiple periods and items. *Computers & Operations Research*, 71, 137-148.
- Çelebi, D. (2015). Inventory control in a centralized distribution network using genetic algorithms: A case study. *Computers & Industrial Engineering*, 87, 532-539.
- Chaharsooghi, S. K., & Heydari, J. (2010). Supply chain coordination for the joint determination of order quantity and reorder point using credit option. *European Journal of Operational Research*, 204(1), 86-95.
- Chakraborty, R., & Hasin, M. (2013). Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach. *International Journal of Industrial Engineering Computations*, 4(1), 1-12.
- Chakraborty, N., Mondal, S., & Maiti, M. (2013). A deteriorating multi-item inventory model with price discount and variable demands via fuzzy logic under resource constraints. *Computers & Industrial Engineering*, 66(4), 976-987.
- Chambari, A., Abbas Najafi, A., Rahmati, S. H. A., & Karimi, A. (2013). An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies. *Reliability Engineering & System Safety*, 119, 158-164.

- Chan, F. T., Kumar, N., Tiwari, M., Lau, H., & Choy, K. (2008). Global supplier selection: a fuzzy-AHP approach. *International Journal of Production Research*, 46(14), 3825-3857.
- Chang, C.-T., & Chou, H.-C. (2013). A coordination system for seasonal demand problems in the supply chain. *Applied Mathematical Modelling*, 37(6), 3674-3686.
- Chang, C., & Chang, S. (2001). On the inventory model with variable lead time and price-quantity discount. *Journal of the Operational Research Society*, 1151-1158.
- Chen, H.-C., Wee, H.-M., Hsieh, Y.-H., & Jou, Y.-T. (2008). *A newsboy inventory model with recyclable and defective items*. Paper presented at the Machine Learning and Cybernetics, 2008 International Conference on.
- Chen, H.D., Wang, Z.P., & Chen, Y. (2015). An integrated model for lot sizing with supplier selection and quantity discounts. *ICIC express letters. Part B, Applications: an international journal of research and surveys*, 6(1), 215-219.
- Chen, L., Chen, Y. F., & Pang, Z. (2011). Dynamic pricing and inventory control in a make-to-stock queue with information on the production status. *Automation Science and Engineering, IEEE Transactions on*, 8(2), 361-373.
- Chen, S.-P., & Ho, Y.-H. (2013). Optimal inventory policy for the fuzzy newsboy problem with quantity discounts. *Information Sciences: an International Journal*, 228, 75-89.
- Chen, Y.-R., & Dye, C.-Y. (2013). Application of particle swarm optimisation for solving deteriorating inventory model with fluctuating demand and controllable deterioration rate. *International Journal of Systems Science*, 44(6), 1026-1039.
- Cheng, C.-Y., Chen, Y.-Y., Chen, T.-L., & Jung-Woon Yoo, J. (2015). Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem. *International Journal of Production Economics*, 170, 805-814.
- Cheng, H., Huang, W., Zhou, Q., & Cai, J. (2013). Solving fuzzy multi-objective linear programming problems using deviation degree measures and weighted max-min method. *Applied Mathematical Modelling*, 37(10), 6855-6869.
- Chiang, C. (2003). Optimal replenishment for a periodic review inventory system with two supply modes. *European Journal of Operational Research*, 149(1), 229-244.
- Choi, T.-M., Chiu, C.-H., & Fu, P.-L. (2011). Periodic review multiperiod inventory control under a mean-variance optimization objective. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(4), 678-682.
- Choudhary, D., & Shankar, R. (2011). Modeling and analysis of single item multiperiod procurement lot-sizing problem considering rejections and late deliveries. *Computers & Industrial Engineering*, 61(4), 1318-1323.

- Coello Coello, C. A., & Lechuga, M. S. (2002). *MOPSO: A proposal for multiple objective particle swarm optimization*. Paper presented at the Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on.
- Cong, Y., Wang, J., & Li, X. (2016). Traffic Flow Forecasting by a Least Squares Support Vector Machine with a Fruit Fly Optimization Algorithm. *Procedia Engineering*, 137, 59-68.
- Cooper, L. (1963). Location-allocation problems. *Operations Research*, 11(3), 331-343.
- Costantino, F., Di Gravio, G., Shaban, A., & Tronci, M. (2016). Smoothing inventory decision rules in seasonal supply chains. *Expert Systems with Applications*, 44, 304-319.
- Dabiri, N., J. Tarokh, M., & Alinaghian, M. (2017). New mathematical model for the bi-objective inventory routing problem with a step cost function: A multi-objective particle swarm optimization solution approach. *Applied Mathematical Modelling*, 49, 302-318.
- Dai, H., Liu, A., Lu, J., Dai, S., Wu, X., & Sun, Y. (2015). Optimization about the layout of IMUs in large ship based on fruit fly optimization algorithm. *Optik - International Journal for Light and Electron Optics*, 126(4), 490-493.
- Dai, H., Zhao, G., Lu, J., & Dai, S. (2014). Comment and Improvement on "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example". *Knowledge-Based Systems*, 59, 159-160.
- Das, B., & Maiti, M. (2007). An application of bi-level newsboy problem in two substitutable items under capital cost. *Applied Mathematics and Computation*, 190(1), 410-422.
- Das, K., Roy, T., & Maiti, M. (2000). Multi-item inventory model with quantity-dependent inventory costs and demand-dependent unit cost under imprecise objective and restrictions: a geometric programming approach. *Production Planning & Control*, 11(8), 781-788.
- Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2015). A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research*, 55, 167-184.
- Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2016). An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95, 95-123.
- De, S. K., & Sana, S. S. (2013). Fuzzy order quantity inventory model with fuzzy shortage quantity and fuzzy promotional index. *Economic Modelling*, 31, 351-358.
- Dey, J. K., Mondal, S. K., & Maiti, M. (2008). Two storage inventory problem with dynamic demand and interval valued lead-time over finite time horizon under inflation and time-value of money. *European Journal of Operational Research*, 185(1), 170-194.

- Diabat, A., Abdallah, T., & Henschel, A. (2013). A closed-loop location-inventory problem with spare parts consideration. *Computers & Operations Research*, 54, 245-256.
- Diabat, A., Aouam, T., & Ozsen, L. (2009). An evolutionary programming approach for solving the capacitated facility location problem with risk pooling. *International Journal of Applied Decision Sciences*, 2(4), 389-405.
- Diabat, A., Richard, J.P., & Codrington, C. W. (2013). A Lagrangian relaxation approach to simultaneous strategic and tactical planning in supply chain design. *Annals of Operations Research*, 203(1), 55-80.
- Diewert, W. E., Finkel, Y., & Artsev, Y. (2009). Empirical Evidence on the Treatment of Seasonal Products: The Israeli CPI Experience. *chapter*, 4, 53-78.
- Dorigo, M., & Birattari, M. (2010). Ant colony optimization *Encyclopedia of Machine Learning* (pp. 36-39): Springer.
- Duda, J. (2017). A hybrid genetic algorithm and variable neighborhood search for multi-family capacitated lot-sizing problem. *Electronic Notes in Discrete Mathematics*, 58, 103-110.
- Dutta, P., & Chakraborty, D. (2008). Single-Period Inventory Models in Fuzzy Environment: A Decision-Maker's Attitudinal Approach. *International Journal of Applied Mathematics and Statistics*, 13(D08), 24-35.
- Dye, C.-Y. (2012). A finite horizon deteriorating inventory model with two-phase pricing and time-varying demand and cost under trade credit financing using particle swarm optimization. *Swarm and Evolutionary Computation*, 5, 37-53.
- Dye, C.-Y., & Ouyang, L.-Y. (2011). A particle swarm optimization for solving joint pricing and lot-sizing problem with fluctuating demand and trade credit financing. *Computers & Industrial Engineering*, 60(1), 127-137.
- Eberhart, R., & Kennedy, J. (1995). *A new optimizer using particle swarm theory*. Paper presented at the Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on.
- Ebrahimipour, V., Najjarbashi, A., & Sheikhalishahi, M. (2015). Multi-objective modeling for preventive maintenance scheduling in a multiple production line. *Journal of Intelligent Manufacturing*, 26(1), 111-122.
- Eppen, G. D., & Iyer, A. V. (1997). Backup agreements in fashion buying—the value of upstream flexibility. *Management Science*, 43(11), 1469-1484.
- Esfandiari, N., & Seifbarghy, M. (2013). Modeling a stochastic multi-objective supplier quota allocation problem with price-dependent ordering. *Applied Mathematical Modelling*, 37(8), 5790-5800.
- Fattahi, P., Hajipour, V., & Nobari, A. (2015). A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms. *Applied Soft Computing*, 32, 211-223.

- Ganesh, M., Raghunathan, S., & Rajendran, C. (2013). The value of information sharing in a multi-product, multi-level supply chain: Impact of product substitution, demand correlation, and partial information sharing. *Decision Support Systems*, 58, 79-94.
- Gebennini, E., Gamberini, R., & Manzini, R. (2009). An integrated production–distribution model for the dynamic location and allocation problem with safety stock optimization. *International Journal of Production Economics*, 122(1), 286-304.
- Ghiami, Y., Williams, T., & Wu, Y. (2013). A two-echelon inventory model for a deteriorating item with stock-dependent demand, partial backlogging and capacity constraints. *European Journal of Operational Research*, 231(3), 587-597.
- Ghodratnama, A., Tavakkoli-Moghaddam, R., & Azaron, A. (2015). Robust and fuzzy goal programming optimization approaches for a novel multi-objective hub location-allocation problem: A supply chain overview. *Applied Soft Computing*, 37, 255-276.
- Gholamian, N., Mahdavi, I., & Tavakkoli-Moghaddam, R. (2015). Multi-objective multi-product multi-site aggregate production planning in a supply chain under uncertainty: fuzzy multi-objective optimisation. *International Journal of Computer Integrated Manufacturing*, 29(2), 149-165.
- Gholamian, N., Mahdavi, I., Tavakkoli-Moghaddam, R., & Mahdavi-Amiri, N. (2015). Comprehensive fuzzy multi-objective multi-product multi-site aggregate production planning decisions in a supply chain under uncertainty. *Applied Soft Computing*, 37, 585-607.
- Gigras, Y., & Gupta, K. (2012). Artificial Intelligence in Robot Path Planning. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, 2231-2307.
- Guchhait, P., Maiti, M. K., & Maiti, M. (2013). A production inventory model with fuzzy production and demand using fuzzy differential equation: an interval compared genetic algorithm approach. *Engineering Applications of Artificial Intelligence*, 26(2), 766-778.
- Gümüş, D. B., Özcan, E., & Atkin, J. (2016). An analysis of the Taguchi method for tuning a memetic algorithm with reduced computational time budget. *Computer and Information Sciences*. 12-20.
- Hajipour, V., & Pasandideh, S. H. R. (2012). Proposing an adaptive particle swarm optimization for a novel bi-objective queuing facility location model. *Economic Computation and Economic Cybernetics Studies and Research*, 46(3), 223-240.
- Hajipour, V., Rahmati, S. H. A., Pasandideh, S. H. R., & Niaki, S. T. A. (2014). A multi-objective harmony search algorithm to optimize multi-server location–allocation problem in congested systems. *Computers & Industrial Engineering*, 72, 187-197.
- Handfield, R., Warsing, D., & Wu, X. (2009). (Q,r) Inventory policies in a fuzzy uncertain supply chain environment. *European Journal of Operational Research*, 197(2), 609-619.

- Harris, I., Mumford, C. L., & Naim, M. M. (2014). A hybrid multi-objective approach to capacitated facility location with flexible store allocation for green logistics modeling. *Transportation Research Part E: Logistics and Transportation Review*, 66, 1-22.
- Hiassat, A., Diabat, A., & Rahwan, I. (2017). A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of Manufacturing Systems*, 42, 93-103.
- Hnaïen, F., Dolgui, A., & Wu, D. D. (2015). Single-period inventory model for one-level assembly system with stochastic lead times and demand. *International Journal of Production Research*, 1-18.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press.
- Hosseini-ninezhad, S. J., Jabalameli, M. S., & Naini, S. G. J. (2013). A fuzzy algorithm for continuous capacitated location allocation model with risk consideration. *Applied Mathematical Modelling*, 38(3), 983-1000.
- Huang, K. L., Kuo, C.W., & Lu, M.L. (2014). Wholesale price rebate vs. capacity expansion: The optimal strategy for seasonal products in a supply chain. *European Journal of Operational Research*, 234(1), 77-85.
- Huang, S.H., & Lin, P.C. (2010). A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 46(5), 598-611.
- Hwang, C.-L., & Masud, A. S. M. (1979). *Multiple objective decision making methods and applications*, 164, Springer
- Janakiraman, G., Park, S. J., Seshadri, S., & Wu, Q. (2013). New results on the newsvendor model and the multi-period inventory model with backordering. *Operations Research Letters*, 41(4), 373-376.
- Jiao, W., Zhang, J.-L., & Yan, H. (2017). The stochastic lot-sizing problem with quantity discounts. *Computers & Operations Research*, 80, 1-10.
- Kao, C., & Hsu, W.-K. (2002). Lot size-reorder point inventory model with fuzzy demands. *Computers & Mathematics with Applications*, 43(10), 1291-1302.
- Karakaya, S., & Bakal, İ. S. (2013). Joint quantity flexibility for multiple products in a decentralized supply chain. *Computers & Industrial Engineering*, 64(2), 696-707.
- Karimi-Nasab, M., Modarres, M., & Seyedhoseini, S. M. (2015). A self-adaptive PSO for joint lot sizing and job shop scheduling with compressible process times. *Applied Soft Computing*, 27, 137-147.
- Karmaker, C., & Saha, M. (2015). Optimization of warehouse location through fuzzy multi-criteria decision making methods. *Decision Science Letters*, 4(3), 315-334.

- Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization*. Paper presented at the Proceedings of IEEE international conference on neural networks.
- Khalili-Damghani, K., & Shahrokh, A. (2014). Solving a New Multi-Period Multi-Objective Multi-Product Aggregate Production Planning Problem Using Fuzzy Goal Programming. *Industrial Engineering & Management Systems*, 13(4), 369-382.
- Khan, M., Jaber, M. Y., & Ahmad, A.-R. (2014). An integrated supply chain model with errors in quality inspection and learning in production. *Omega*, 42(1), 16-24.
- Kim, I. Y., & de Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31(2), 105-116. doi:10.1007/s00158-005-0557-6
- Kim, J.U., & Kim, Y.D. (2000). A Lagrangian relaxation approach to multi-period inventory/distribution planning. *Journal of the Operational Research Society*, 51(3), 364-370.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Kundu, A., Guchhait, P., Pramanik, P., Kumar Maiti, M., & Maiti, M. (2017). A production inventory model with price discounted fuzzy demand using an interval compared hybrid algorithm. *Swarm and Evolutionary Computation*, 34, 1-17.
- Latha Shankar, B., Basavarajappa, S., Chen, J. C., & Kadadevaramath, R. S. (2012). Location and allocation decisions for multi-echelon supply chain network- A multi-objective evolutionary approach. *Expert Systems with Applications*, 40(2), 551-562.
- Lee, A. H., & Kang, H.-Y. (2008). A mixed 0-1 integer programming for inventory model: a case study of TFT-LCD manufacturing company in Taiwan. *Kybernetes*, 37(1), 66-82.
- Lee, S.-h. (2017). A fuzzy multi-objective programming approach for determination of resilient supply portfolio under supply failure risks. *Journal of Purchasing and Supply Management*, 23(3), 211-220.
- Li, H.z., Guo, S., Li, C.-j., & Sun, J.-q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowledge-Based Systems*, 37, 378-387.
- Li, J., & Liu, L. (2006). Supply chain coordination with quantity discount policy. *International Journal of Production Economics*, 101(1), 89-98.
- Li, K., Chen, B., Sivakumar, A. I., & Wu, Y. (2014). An inventory–routing problem with the objective of travel time minimization. *European Journal of Operational Research*, 236(3), 936-945.
- Li, Y., Tao, Y., & Wang, F. (2012). An effective approach to multi-item capacitated dynamic lot-sizing problems. *International Journal of Production Research*, 50(19), 5348-5362.

- Lin, T.-Y. (2013). Coordination policy for a two-stage supply chain considering quantity discounts and overlapped delivery with imperfect quality. *Computers & Industrial Engineering*, 66(1), 53-62.
- Liu, J., & Luo, C. (2009). *Inventory Control with Return Policy Under Conditional Value-at-Risk Criterion*. Paper presented at the Management and Service Science, 2009. MASS'09. International Conference on.
- Liu, H., Zhang, J., Zhou, C., & Ru, Y. (2017). Optimal purchase and inventory retrieval policies for perishable seasonal agricultural products. *Omega*. doi:http://dx.doi.org/10.1016/j.omega.2017.08.006
- Logendran, R., & Terrell, M. P. (1988). Uncapacitated plant location-allocation problems with price sensitive stochastic demands. *Computers & Operations Research*, 15(2), 189-198.
- Luciano, E., Peccati, L., & Cifarelli, D. M. (2003). VaR as a risk measure for multiperiod static inventory models. *International Journal of Production Economics*, 81, 375-384.
- Ma, P., Han, L., Liu, H., & Zhang, H. (2017). Fuzzy Multi-objective Optimization of EMT Based on the Minimum Average Weighted Deviation Algorithm. *Energy Procedia*, 105, 2409-2414.
- Maiti, A., & Maiti, M. (2008). Discounted multi-item inventory model via genetic algorithm with roulette wheel selection, arithmetic crossover and uniform mutation in constraints bounded domains. *International Journal of Computer Mathematics*, 85(9), 1341-1353.
- Maity, K. (2011). Possibility and necessity representations of fuzzy inequality and its application to two warehouse production-inventory problem. *Applied Mathematical Modelling*, 35(3), 1252-1263.
- Maity, K., & Maiti, M. (2008). A numerical approach to a multi-objective optimal inventory control problem for deteriorating multi-items under fuzzy inflation and discounting. *Computers & Mathematics with Applications*, 55(8), 1794-1807.
- Majidi, M., Nojavan, S., Nourani Esfetanaj, N., Najafi-Ghalelou, A., & Zare, K. (2017). A multi-objective model for optimal operation of a battery/PV/fuel cell/grid hybrid energy system using weighted sum technique and fuzzy satisfying approach considering responsible load management. *Solar Energy*, 144, 79-89.
- Malakooti, B. (2013). *Operations and production systems with multiple objectives*: John Wiley & Sons.
- Masmoudi, O., Yalaoui, A., Ouazene, Y., & Chehade, H. (2016). Multi-item capacitated lot-sizing problem in a flow-shop system with energy consideration. *IFAC-PapersOnLine*, 49(12), 301-306.
- Matsuyama, K. (2006). The multi-period newsboy problem. *European Journal of Operational Research*, 171(1), 170-188.
- Menlo, W. (2007). Logistics industry.

- Mentzer, J. T., DeWitt, W., Keebler, J. S., Min, S., Nix, N. W., Smith, C. D., & Zacharia, Z. G. (2001). Defining supply chain management. *Journal of Business logistics*, 22(2), 1-25.
- Mirzapour Al-e-hashem, S., Baboli, A., & Sazvar, Z. (2013). A stochastic aggregate production planning model in a green supply chain: Considering flexible lead times, nonlinear purchase and shortage cost functions. *European Journal of Operational Research*, 230(1), 26-41.
- Mirzapour Al-e-hashem, S., & Rekik, Y. (2014). Multi-product multi-period Inventory Routing Problem with a transshipment option: A green approach. *International Journal of Production Economics*, 157, 80-88.
- Moattar Hussein, Z., Karimi, B., Moattar Hussein, S. M., & Ghodsipour, S. H. (2014). Multi-objective integrated production distribution planning concerning manufacturing partners. *International Journal of Computer Integrated Manufacturing*, 28(12) 1313-1330.
- Mogale, D. G., Dolgui, A., Kandhway, R., Kumar, S. K., & Tiwari, M. K. (2017). A multi-period inventory transportation model for tactical planning of food grain supply chain. *Computers & Industrial Engineering*, 110, 379-394.
- Mohammadi, M. (2015). *Metaheuristic algorithms for solving lot-sizing and scheduling problems in single and multi-plant environments*. University of Malaya.
- Mohammadi, M., Musa, S. N., & Bahreininejad, A. (2015). Optimization of economic lot scheduling problem with backordering and shelf-life considerations using calibrated metaheuristic algorithms. *Applied Mathematics and Computation*, 251, 404-422.
- Mohebbi, E., & Posner, M. J. (2002). Multiple replenishment orders in a continuous-review inventory system with lost sales. *Operations Research Letters*, 30(2), 117-129.
- Mousavi, S. M., Alikar, N., & Niaki, S. T. A. (2016). An improved fruit fly optimization algorithm to solve the homogeneous fuzzy series-parallel redundancy allocation problem under discount strategies. *Soft Computing*, 20(6), 2281-2307.
- Mousavi, S. M., Alikar, N., Niaki, S. T. A., & Bahreininejad, A. (2015). Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies. *Applied Mathematical Modelling*, 39(22), 6968-6989.
- Mousavi, S. M., Alikar, N., Tavana, M., & Di Caprio, D. (2017). An improved particle swarm optimization model for solving homogeneous discounted series-parallel redundancy allocation problems. *Journal of Intelligent Manufacturing*, 1-20. DOI. 10.1007/s10845-017-1311-9.
- Mousavi, S. M., Bahreininejad, A., Musa, S. N., & Yusof, F. (2014). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. *Journal of Intelligent Manufacturing*, 28, 191-206.

- Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Alikar, N. (2014). A multi-product multi-period inventory control problem under inflation and discount: a parameter-tuned particle swarm optimization algorithm. *The International Journal of Advanced Manufacturing Technology*, 70(9-12), 1739-1756.
- Mousavi, S. M., Hajipour, V., Niaki, S. T. A., & Alikar, N. (2013). Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: Two calibrated meta-heuristic algorithms. *Applied Mathematical Modelling*, 37(4), 2241-2256.
- Mousavi, S. M., & Niaki, S. T. A. (2013). Capacitated location allocation problem with stochastic location and fuzzy demand: A hybrid algorithm. *Applied Mathematical Modelling*, 37(7), 5109-5119.
- Mousavi, S. M., Niaki, S. T. A., Bahreininejad, A., & Musa, S. N. (2014). Multi-item multiperiodic inventory control problem with variable demand and discounts: a particle swarm optimization algorithm. *The Scientific World Journal*, 2014, 1-16.
- Mousavi, S. M., Niaki, S. T. A., Mehdizadeh, E., & Tavarroth, M. R. (2013). The capacitated multi-facility location-allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms. *International Journal of Systems Science*, 44(10), 1897-1912.
- Mousavi, S. M., Sadeghi, J., Niaki, S. T. A., Alikar, N., Bahreininejad, A., & Metselaar, H. S. C. (2014b). Two parameter-tuned meta-heuristics for a discounted inventory control problem in a fuzzy environment. *Information Sciences*, 276, 42-62.
- Mousavi, S. M., Sadeghi, J., Niaki, S. T. A., & Tavana, M. (2016). A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NREGA, and MOPSO. *Applied Soft Computing*, 43, 57-72.
- Mousavi, S. M., Tavana, M., Alikar, N., & Zandieh, M. (2017). A tuned hybrid intelligent fruit fly optimization algorithm for fuzzy rule generation and classification. *Neural Computing and Applications*. doi:10.1007/s00521-017-3115-4
- Naka, S., Genji, T., Yura, T., & Fukuyama, Y. (2001). *Practical distribution state estimation using hybrid particle swarm optimization*. Paper presented at the Power Engineering Society Winter Meeting, 2001. IEEE.
- Noblesse, A. M., Boute, R. N., Lambrecht, M. R., & Van Houdt, B. (2014). Lot sizing and lead time decisions in production/inventory systems. *International Journal of Production Economics*, 155, 351-360.
- Omar, O. D. (2013). Supply chain coordination using revenue-dependent revenue sharing contracts. *Omega*, 41(4), 780-796.
- Padmanabhan, G., & Vrat, P. (1990). Analysis of multi-item inventory systems under resource constraints: A non-linear goal programming approach. *Engineering Costs and Production Economics*, 20(2), 121-127.
- Pan, W.-T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26, 69-74.

- Panda, S., Banerjee, K., & Basu, M. (2005). Determination of EOQ of multi-item inventory problems through nonlinear goal programming with penalty function. *Asia-Pacific Journal of Operational Research*, 22(04), 539-553.
- Parhizkari, M., Amirib, M., & Mousakhani, M. (2013). A multiple criteria decision making technique for supplier selection and inventory management strategy: A case of multi-product and multi-supplier problem. *Decision Science Letters*, 2(3), 185-190.
- Park, K., & Kyung, G. (2013). Optimization of total inventory cost and order fill rate in a supply chain using PSO. *The International Journal of Advanced Manufacturing Technology*, 70(9-12), 1533-1541.
- Park, Y.-B., Yoo, J.-S., & Park, H.-S. (2016). A genetic algorithm for the vendor-managed inventory routing problem with lost sales. *Expert Systems with Applications*, 53, 149-159.
- Parsopoulos, K. E., Konstantaras, I., & Skouri, K. (2015). Metaheuristic optimization for the Single-Item Dynamic Lot Sizing problem with returns and remanufacturing. *Computers & Industrial Engineering*, 83, 307-315.
- Pasandideh, S. H. R., Niaki, S. T. A., & Far, M. H. (2014). Optimization of vendor managed inventory of multiproduct EPQ model with multiple constraints using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 71(1-4), 365-376.
- Pasandideh, S. H. R., Niaki, S. T. A., & Hajipour, V. (2013). A multi-objective facility location model with batch arrivals: two parameter-tuned metaheuristic algorithms. *Journal of Intelligent Manufacturing*, 24(2), 331-348.
- Pasandideh, S. H. R., Niaki, S. T. A., & Mousavi, S. M. (2013). Two metaheuristics to solve a multi-item multiperiod inventory control problem under storage constraint and discounts. *The International Journal of Advanced Manufacturing Technology*, 69(5-8), 1671-1684.
- Paul, S. K., Sarker, R., & Essam, D. (2013). A disruption recovery model in a production-inventory system with demand uncertainty and process reliability *Computer Information Systems and Industrial Management* (pp. 511-522): Springer.
- Peace, G. Taguchi Methods, A Hands On Approach, 1993: Addison-Wesley Publishing Company, Reading, MA, USA.
- Peng, Y., Dong, M., & Han, G. (2010). *Low price guarantee in inventory systems coordinated with pricing strategy*. Paper presented at the Service Operations and Logistics and Informatics (SOLI), 2010 IEEE International Conference on.
- Phadke, M. S. (1995). *Quality engineering using robust design*: Prentice Hall PTR.
- Priyan, S., Palanivel, M., & Uthayakumar, R. (2014). Mathematical modeling for EOQ inventory system with advance payment and fuzzy Parameters. *International Journal of Supply and Operations Management*, 1(3), 260.
- Rahmati, S. H. A., Ahmadi, A., Sharifi, M., & Chambari, A. (2014). A multi-objective model for facility location-allocation problem with immobile

- servers within queuing framework. *Computers & Industrial Engineering*, 74, 1-10.
- Rahmati, S. H. A., Hajipour, V., & Niaki, S. T. A. (2013). A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. *Applied Soft Computing*, 13(4), 1728-1740.
- Ramachandran, B., & Thomas Bellarmine, G. (2014). Improving observability using optimal placement of phasor measurement units. *International Journal of Electrical Power & Energy Systems*, 56, 55-63.
- Ramezani, M., Kimiagari, A. M., Karimi, B., & Hejazi, T. H. (2014). Closed-loop supply chain network design under a fuzzy environment. *Knowledge-Based Systems*, 59, 108-120.
- Rayat, F., Musavi, M., & Bozorgi-Amiri, A. (2017). Bi-objective reliable location-inventory-routing problem with partial backordering under disruption risks: A modified AMOSA approach. *Applied Soft Computing*, 59, 622-643.
- Roach, B. (2005). Origin of the Economic Order Quantity formula; transcription or transformation? *Management decision*, 43(9), 1262-1268.
- Rodriguez, M. A., Vecchiotti, A. R., Harjunkski, I., & Grossmann, I. E. (2014). Optimal supply chain design and management over a multi-period horizon under demand uncertainty. Part I: MINLP and MILP models. *Computers & Chemical Engineering*, 62, 194-210.
- Roosbeh Nia, A., Hemmati Far, M., & Akhavan Niaki, S. T. (2013). A fuzzy vendor managed inventory of multi-item economic order quantity model under shortage: An ant colony optimization algorithm. *International Journal of Production Economics*, 155, 259-271.
- Roosbeh Nia, A., Hemmati Far, M., & Niaki, S. T. A. (2015). A hybrid genetic and imperialist competitive algorithm for green vendor managed inventory of multi-item multi-constraint EOQ model under shortage. *Applied Soft Computing*, 30, 353-364.
- Ross, P. J. P. J. (1996). *Taguchi techniques for quality engineering: loss function, orthogonal experiments, parameter and tolerance design*.
- Sadeghi, J., Mousavi, S. M., Niaki, S. T. A., & Sadeghi, S. (2013b). Optimizing a multi-vendor multi-retailer vendor managed inventory problem: Two tuned meta-heuristic algorithms. *Knowledge-Based Systems*, 50, 159-170.
- Sadeghi, J., Mousavi, S. M., Niaki, S. T. A., & Sadeghi, S. (2014). Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm. *Transportation Research Part E: Logistics and Transportation Review*, 70, 274-292.
- Sadeghi, J., & Niaki, S. T. A. (2015). Two parameter tuned multi-objective evolutionary algorithms for a bi-objective vendor managed inventory model with trapezoidal fuzzy demand. *Applied Soft Computing*, 30, 567-576.
- Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014a). A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain

- management: An NSGA-II with tuned parameters. *Computers & Operations Research*, 41, 53-64.
- Sadeghi, J., Sadeghi, S., & Niaki, S. T. A. (2014b). Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm. *Information Sciences*, 272, 126-144.
- Sadi-Nezhad, S., Nahavandi, S., & Nazemi, J. (2011). Periodic and continuous inventory models in the presence of fuzzy costs. *International Journal of Industrial Engineering Computations*, 2(1), 179-192.
- Samal, N. K., & Pratihar, D. K. (2014). Optimization of variable demand fuzzy economic order quantity inventory models without and with backordering. *Computers & Industrial Engineering*, 78, 148-162.
- Sana, S. S., & Chaudhuri, K. (2008). A deterministic EOQ model with delays in payments and price-discount offers. *European Journal of Operational Research*, 184(2), 509-533.
- Saracoglu, I., Topaloglu, S., & Keskindurk, T. (2014). A genetic algorithm approach for multi-product multi-period continuous review inventory models. *Expert Systems with Applications*, 41(18), 8189-8202.
- Sepehri, M. (2011). Cost and inventory benefits of cooperation in multi-period and multi-product supply. *Scientia Iranica*, 18(3), 731-741.
- Shahabi, M., Akbarinasaji, S., Unnikrishnan, A., & James, R. (2013). Integrated Inventory Control and Facility Location Decisions in a Multi-Echelon Supply Chain Network with Hubs. *Networks and Spatial Economics*, 13(4), 497-514.
- Shavandi, H., Mahlooji, H., & Nosrati, N. E. (2012). A constrained multi-product pricing and inventory control problem. *Applied Soft Computing*, 12(8), 2454-2461.
- Shayeghi, H., Shayanfar, H., Jalilzadeh, S., & Safari, A. (2009). A PSO based unified power flow controller for damping of power system oscillations. *Energy conversion and management*, 50(10), 2583-2592.
- Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., . . . Liu, D. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems*, 96, 61-75.
- Sherali, H. D., & Rizzo, T. P. (1991). Unbalanced, capacitated p- median problems on a chain graph with a continuum of link demands. *Networks*, 21(2), 133-163.
- Shi, Y., & Eberhart, R. C. (1999). *Empirical study of particle swarm optimization*. Paper presented at the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on.
- Shi, J., Guo, J. e., & Fung, R. Y. K. (2017). Decision support system for purchasing management of seasonal products: A capital-constrained retailer perspective. *Expert Systems with Applications*, 80, 171-182.

- Shin, H., & Benton, W. (2004). Quantity Discount- Based Inventory Coordination: Effectiveness and Critical Environmental Factors. *Production and Operations Management*, 13(1), 63-76.
- Sifaleras, A., & Konstantaras, I. (2017). Variable neighborhood descent heuristic for solving reverse logistics multi-item dynamic lot-sizing problems. *Computers & Operations Research*, 78, 385-392.
- Silver, E. A., & Moon, I. (2001). Multi-item economic order quantity model with an initial stock of convertible units. *The Engineering Economist*, 46(2), 129-138.
- Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2000). Design and managing the supply chain: Concepts, strategies, and case studies: Boston: Irwin/McGraw-Hill.
- Singh, S., & Saxena, N. (2013). A Closed Loop Supply Chain System with Flexible Manufacturing and Reverse Logistics Operation under Shortages for Deteriorating Items. *Procedia Technology*, 10, 330-339.
- Soleimani, H., & Kannan, G. (2015). A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks. *Applied Mathematical Modelling*, 39(14), 3990-4012.
- Soni, N., & Kumar, T. (2014). Study of various crossover operators in genetic algorithms. *International Journal of Computer Science and Information Technologies*, 5(6), 7235-7238.
- Spekman, R. E., Kamauff Jr, J. W., & Myhr, N. (1998). An empirical investigation into supply chain management: a perspective on partnerships. *Supply Chain Management: An International Journal*, 3(2), 53-67.
- Srivastav, A., & Agrawal, S. (2016). Multi-objective optimization of hybrid backorder inventory model. *Expert Systems with Applications*, 51, 76-84.
- Stojiljković, M. M. (2017). Bi-level multi-objective fuzzy design optimization of energy supply systems aided by problem-specific heuristics. *Energy*. doi.org/10.1016/j.energy.2017.06.037
- Su, T. S. (2017). A fuzzy multi-objective linear programming model for solving remanufacturing planning problems with multiple products and joint components. *Computers & Industrial Engineering*, 110, 242-254.
- Swinney, R., Cachon, G. P., & Netessine, S. (2011). Capacity investment timing by start-ups and established firms in new markets. *Management Science*, 57(4), 763-777.
- Taleizadeh, A. A., Akhavan Niaki, S. T., & Hoseini, V. (2009). Optimizing the multi-product, multi-constraint, bi-objective newsboy problem with discount by a hybrid method of goal programming and genetic algorithm. *Engineering Optimization*, 41(5), 437-457.
- Taleizadeh, A. A., Niaki, S. T. A., & Aryanezhad, M.-B. (2009). A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory control systems with random fuzzy replenishments. *Mathematical and Computer Modelling*, 49(5), 1044-1057.

- Taleizadeh, A. A., Niaki, S. T. A., & Aryanezhad, M.-B. (2010). Optimising multi-product multi-chance-constraint inventory control system with stochastic period lengths and total discount under fuzzy purchasing price and holding costs. *International Journal of Systems Science*, 41(10), 1187-1200.
- Taleizadeh, A. A., Niaki, S. T. A., Aryanezhad, M.-B., & Shafii, N. (2013). A hybrid method of fuzzy simulation and genetic algorithm to optimize constrained inventory control systems with stochastic replenishments and fuzzy demand. *Information Sciences*, 220, 425-441.
- Taleizadeh, A. A., Niaki, S. T. A., Aryanezhad, M.-B., & Tafti, A. F. (2010). A genetic algorithm to optimize multiproduct multiconstraint inventory control systems with stochastic replenishment intervals and discount. *The International Journal of Advanced Manufacturing Technology*, 51(1-4), 311-323.
- Taleizadeh, A. A., Niaki, S. T. A., & Barzinpour, F. (2011). Multiple-buyer multiple-vendor multi-product multi-constraint supply chain problem with stochastic demand and variable lead-time: A harmony search algorithm. *Applied Mathematics and Computation*, 217(22), 9234-9253.
- Taleizadeh, A. A., Niaki, S. T. A., & Hosseini, V. (2008). The multi-product multi-constraint newsboy problem with incremental discount and batch order. *Journal Engineering Optimization*, 1, 110-122.
- Taleizadeh, A. A., Niaki, S. T. A., & Makui, A. (2012). Multiproduct multiple-buyer single-vendor supply chain problem with stochastic demand, variable lead-time, and multi-chance constraint. *Expert Systems with Applications*, 39(5), 5338-5348.
- Taleizadeh, A. A., Niaki, S. T. A., Shafii, N., Meibodi, R. G., & Jabbarzadeh, A. (2010). A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead time and (r, Q) policy in supply chain. *The International Journal of Advanced Manufacturing Technology*, 51(9-12), 1209-1223.
- Tan, K. C. (2001). A framework of supply chain management literature. *European Journal of Purchasing & Supply Management*, 7(1), 39-48.
- Tanksale, A., & Jha, J. K. (2017). Solving multi-region multi-facility inventory allocation and transportation problem: A case of Indian public distribution system. *Computers & Industrial Engineering*, 110, 175-190.
- Thakur, P., & Singh, A. J. (2014). Study of various crossover operators in Genetic Algorithms. *Int. J. Adv. Res. Comput. Sci. Softw. Eng*, 4, 7235-7238.
- Tsai, C.-Y., & Yeh, S.-W. (2008). A multiple objective particle swarm optimization approach for inventory classification. *International Journal of Production Economics*, 114(2), 656-666.
- Tsao, Y.-C., & Lu, J.-C. (2012). A supply chain network design considering transportation cost discounts. *Transportation Research Part E: Logistics and Transportation Review*, 48(2), 401-414.

- Tsao, Y.-C., & Sheen, G.-J. (2012). A multi-item supply chain with credit periods and weight freight cost discounts. *International Journal of Production Economics*, 135(1), 106-115.
- Turban, E., Lee, J. K., King, D., Liang, T. P., & Turban, D. (2009). *Electronic commerce 2010*: Prentice Hall Press.
- Vahdani, B., Soltani, M., Yazdani, M., & Meysam Mousavi, S. (2017). A three level joint location-inventory problem with correlated demand, shortages and periodic review system: Robust meta-heuristics. *Computers & Industrial Engineering*, 109, 113-129.
- Van Mieghem, J. A., & Dada, M. (1999). Price versus production postponement: Capacity and competition. *Management Science*, 45(12), 1639-1649.
- Vital Soto, A., Chowdhury, N. T., Allahyari, M. Z., Azab, A., & Baki, M. F. (2017). Mathematical modeling and hybridized evolutionary LP local search method for lot-sizing with supplier selection, inventory shortage, and quantity discounts. *Computers & Industrial Engineering*, 109, 96-112.
- Wang, D., Qin, Z., & Kar, S. (2015). A novel single-period inventory problem with uncertain random demand and its application. *Applied Mathematics and Computation*, 269, 133-145.
- Wang, K.-J., & Lee, C.-H. (2015). A revised ant algorithm for solving location-allocation problem with risky demand in a multi-echelon supply chain network. *Applied Soft Computing*, 32, 311-321.
- Wang, L., Shi, Y., & Liu, S. (2015). An improved fruit fly optimization algorithm and its application to joint replenishment problems. *Expert Systems with Applications*, 42(9), 4310-4323.
- Wang, L., & Zheng, X.-l. (2017). A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem. *Swarm and Evolutionary Computation*. doi.org/10.1016/j.swevo.2017.06.001.
- Wang, L., Zheng, X.-l., & Wang, S.-y. (2013). A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. *Knowledge-Based Systems*, 48, 17-23.
- Wang, X., & Xu, Y. (2009). Multi-period multi-product inventory control with substitution and an exponential criterion. *International Journal of Revenue Management*, 3(4), 333-353.
- Wang, Y. M., & Yin, H. L. (2013). *Integrated optimization for supply chain with facility disruption*. Paper presented at the Mechatronics and Automation (ICMA), 2013 IEEE International Conference on.
- Wen, M., & Iwamura, K. (2008a). Facility location-allocation problem in random fuzzy environment: Using (α, β) -cost minimization model under the Hurewicz criterion. *Computers & Mathematics with Applications*, 55(4), 704-713.

- Wen, M., & Iwamura, K. (2008b). Fuzzy facility location-allocation problem under the Hurwicz criterion. *European Journal of Operational Research*, 184(2), 627-635.
- Willoughby, K. A., & Uyeno, D. H. (2001). Resolving splits in location/allocation modeling: a heuristic procedure for transit center decisions. *Transportation Research Part E: Logistics and Transportation Review*, 37(1), 71-83.
- Wilson, R. (2005). 16th annual "state of logistics report". Council of supply chain management professionals. *Oak Brook, IL, USA*.
- Wu, L., Zuo, C., & Zhang, H. (2015). A cloud model based fruit fly optimization algorithm. *Knowledge-Based Systems*, 89, 603-617.
- Xiao, G., Jin, X., Chang, Q., Biller, S., Ni, J., & Hu, S. J. (2012). *Performance Analysis and Optimization of Remanufacturing Systems with Stochastic Returns*. Paper presented at the ASME 2012 International Manufacturing Science and Engineering Conference collocated with the 40th North American Manufacturing Research Conference and in participation with the International Conference on Tribology Materials and Processing.
- Yaghin, R. G., Ghomi, S. F., & Torabi, S. A. (2013). A possibilistic multiple objective pricing and lot-sizing model with multiple demand classes. *Fuzzy Sets and Systems*, 231, 26-44.
- Yang, M.-F., & Lin, Y. (2010). Applying the linear particle swarm optimization to a serial multi-echelon inventory model. *Expert Systems with Applications*, 37(3), 2599-2608.
- Yang, Y., Chen, Y., & Zhou, Y. (2014). Coordinating inventory control and pricing strategies under batch ordering. *Operations Research*, 62(1), 25-37.
- Yousefi, O., Aryanezhad, M., Sadjadi, S. J., & Shahin, A. (2012). Developing a multi-objective, multi-item inventory model and three algorithms for its solution. *Journal of Zhejiang University SCIENCE C*, 13(8), 601-612.
- Yu, Y., Li, Y., Li, J., & Gu, X. (2016). Self-adaptive step fruit fly algorithm optimized support vector regression model for dynamic response prediction of magnetorheological elastomer base isolator. *Neurocomputing*, 211, 41-52.
- Yuan, B., & Gallagher, M. (2005). *A hybrid approach to parameter tuning in genetic algorithms*. Paper presented at the Evolutionary Computation, 2005. The 2005 IEEE Congress on.
- Zhang, D. (2013). An integrated production and inventory model for a whole manufacturing supply chain involving reverse logistics with finite horizon period. *Omega*, 41(3), 598-620.
- Zhang, D., Xu, H., & Wu, Y. (2009). Single and multi-period optimal inventory control models with risk-averse constraints. *European Journal of Operational Research*, 199(2), 420-434.
- Zhang, P., & Wang, L. (2013). *An Improved Fruit Fly Optimization Algorithm for Solving Lot-streaming Flow-shop Scheduling Problem*. Paper presented at

the The 3rd International Workshop on Advanced Computational Intelligence and Intelligent Informatics, Shanghai, IWACIII.

- Zhang, T., Zheng, Q. P., Fang, Y., & Zhang, Y. (2015). Multi-level inventory matching and order planning under the hybrid Make-To-Order/Make-To-Stock production environment for steel plants via Particle Swarm Optimization. *Computers & Industrial Engineering*, 87, 238-249.
- Zheng, X.-l., & Wang, L. (2016). A two-stage adaptive fruit fly optimization algorithm for unrelated parallel machine scheduling problem with additional resource constraints. *Expert Systems with Applications*, 65, 28-39.
- Zheng, X.-l., Wang, L., & Wang, S.-y. (2013). A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. *Knowledge-Based Systems*, 57, 95-103.
- Zhou, J. (2000). *Uncapacitated facility layout problem with stochastic demands*. Paper presented at the Proceedings of the Sixth National Conference of Operations Research Society of China.
- Zhou, J., & Liu, B. (2003). New stochastic models for capacitated location-allocation problem. *Computers & Industrial Engineering*, 45(1), 111-125.
- Zhou, J., & Liu, B. (2007). Modeling capacitated location-allocation problem with fuzzy demands. *Computers & Industrial Engineering*, 53(3), 454-468.
- Zhou, W., Lee, C. Y., & Wu, D. (2011). Optimal control of a capacitated inventory system with multiple demand classes. *Naval Research Logistics (NRL)*, 58(1), 43-58.
- Zouadi, T., Yalaoui, A., Reghioui, M., & El Kadiri, K. E. (2013). *Lot-sizing for production planning in a recovery system with returns*. Paper presented at the Industrial Engineering and Systems Management (IESM), Proceedings of 2013 International Conference on.

LIST OF PUBLICATIONS AND PAPERS PRESENTED

1. Seyed Mohsen Mousavi, Seyed Taghi Akhavan Niaki, Ardeshir Bahreininejad, Siti Nurmaya Musa; *Multi-Item Multiperiodic Inventory Control Problem with Variable Demand and Discounts: A Particle Swarm Optimization Algorithm*; The Scientific World Journal. Volume 2014 (2014), Article ID 136047, 16 pages (Q1).
2. Seyed Mohsen Mousavi, Ardeshir Bahreininejad, Siti Nurmaya Musa, Farazila Yusof; *A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network*; Journal of Intelligent Manufacturing. January 2017, Volume 28, Issue 1, pp 191–206. (Q2).
3. Seyed Mohsen Mousavi, Najmeh Alikar, Seyed Taghi Akhavan Niaki, Ardeshir Bahreininejad; *Optimizing a location allocation-inventory problem in a two-echelon supply chain network: A modified fruit fly optimization algorithm*; Computers & Industrial Engineering. September 2015, Volume 87, pp 543–560. (Q1).
4. Seyed Mohsen Mousavi, Javad Sadeghi, Seyed Taghi Akhavan Niaki, Najmeh Alikar, Ardeshir Bahreininejad, Hendrik Simon Cornelis Metselaar; *Two parameter-tuned meta-heuristics for a discounted inventory control problem in a fuzzy environment*; Information Sciences. Volume 276, 20 August 2014, Pages 42–62. (Q1).
5. Seyed Mohsen Mousavi, Javad Sadeghi, Seyed Taghi Akhavan Niaki, Madjid Tavana; *A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NPGA, and MOPSO*; Applied Soft Computing. Volume 43, 2016, Pages 57–72. (Q1).
6. Seyed Mohsen Mousavi, Vahid Hajipour, Seyed Taghi Akhavan Niaki, Najmeh Alikar; *A multi-product multi-period inventory control problem under inflation and discount: a parameter-tuned particle swarm optimization algorithm*; The International Journal of Advanced Manufacturing Technology. Volume 70, 20 August 2014, Pages 1739–1756. (Q1).

7. Seyed Hamidreza Pasandideh, Seyed Taghi Akhavan Niaki, Seyed Mohsen Mousavi; *Two metaheuristics to solve a multi-item multiperiod inventory control problem under storage constraint and discounts*; *The International Journal of Advanced Manufacturing Technology*. Volume 69, 20 August 2014, Pages 1671-1684. (Q1).
8. Javad Sadeghi, Seyed Mohsen Mousavui, Seyed Taghi Akhavan Niaki; *Optimizing an inventory model with fuzzy demand, backordering, and discount using a hybrid imperialist competitive algorithm*; *Applied Mathematical Modelling*. Volume 40, 2016, Pages 7318-7335. (Q1).
9. Seyed Mohsen Mousavui, Najmeh Alikar, Najmeh Alikar, Ardeshir Bahreininejad; *Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies*; *Applied Mathematical Modelling*. Volume 39, 2015, Pages 6968-6989. (Q1).
10. Seyed Mohsen Mousavi, Madjid Tavana, Seyed Taghi Akhavan Niaki, Mostafa Zandieh; *A tuned hybrid intelligent fruit fly optimization algorithm for fuzzy rule generation and classification*; *Neural Computing and Applications*. 2017, Pages 1-13. (Q2).
11. Najmeh Alikar, Seyed Mohsen Mousavi, Madjid Tavana, Seyed Taghi Akhavan Niaki, Mostafa Zandieh; *A Multi-Objective Multi-State Series-Parallel Redundancy Allocation Model Using Tuned Meta-Heuristic Algorithms*; *International Journal of Systems Science: Operations & Logistics*. 2017, Pages 1-22. (Q2).

Conference proceeding

1. Seyed Mohsen Mousavi, Najmeh Alikar, Siti Nurmaya Musa, Seyed Taghi Akhavan Niaki, *A Multi-product Multi-period location-inventory for a supply chain network under inflation and time value of money*; 2nd International Conference on Industrial and Systems Engineering (ICISE), Ferdowsi University of Mashhad, Bamonar St., Mashhad, Iran (2016).

Appendix A: THE DATA SHEET COLLECTED FROM THE COMPANY

صنایع چوب ملامین عالی کار

تولید کننده روکش هایگلاس تحت لیسانس آلمان



تاریخ: ۹۵/۵/۲۴

شماره:

پوست:

ردیف	نام کالا	کد کالا	واحد	تعداد	دوره	هزینه کسود	هزینه مشتری	هزینه خرید	هزینه نگهداری	هزینه سفارش	فضای انبار
۱	خام MDF	۰۱۰۱۲۸۱۶	ورق	۱۱۰	دوره اول	۵۰۰	۸۰	۵۲۰۰۰	۱۰۰۰	۳۰۰۰	۵۰۰
۲	خام MDF	۰۱۰۱۲۸۱۶	ورق	۱۱۰	دوره دوم	۶۰۰	۹۰	۵۲۰۰۰	۱۰۰۰	۳۰۰۰	۵۰۰
۳	خام MDF	۰۱۰۱۲۸۱۶	ورق	۱۸۰	دوره سوم	۸۰۰	۱۱۰	۵۲۰۰۰	۱۰۰۰	۳۰۰۰	۵۰۰
۴	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۵۰	دوره اول	۵۰۰	۸۰	۳۰۰۰۰	۱۰۰	۳۰۰	۱۰۰
۵	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۵۰	دوره دوم	۶۰۰	۹۰	۳۰۰۰۰	۱۰۰	۳۰۰	۱۰۰
۶	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۶۰	دوره سوم	۸۰۰	۱۱۰	۳۰۰۰۰	۱۰۰	۳۰۰	۱۰۰
۷	رول هایگلاس	۲۰۱۲۴۱۶	متری	۱۲۰	دوره اول	۳۲	۳۰۰	۸۰۰۰۰۰	۵۰۰	۴۰۰۰	۲۰۰
۸	رول هایگلاس	۲۰۱۲۴۱۶	متری	۱۲۰	دوره دوم	۳۲	۴۰۰	۸۰۰۰۰۰	۵۰۰	۴۰۰۰	۲۰۰
۹	رول هایگلاس	۲۰۱۲۴۱۶	متری	۱۲۰	دوره سوم	۵۰	۶۰۰	۸۰۰۰۰۰	۵۰۰	۴۰۰۰	۲۰۰
۱۰	رول پی وی سی	۲۰۱۲۴۱۵	متری	۱۲۰	دوره اول	۳۲	۱۰۰	۱۵۰۰۰۰	۵۰۰	۱۰۰۰	۱۰۰
۱۱	رول پی وی سی	۲۰۱۲۴۱۵	متری	۱۲۰	دوره دوم	۳۲	۲۰۰	۱۵۰۰۰۰	۵۰۰	۱۰۰۰	۱۰۰
۱۲	رول پی وی سی	۲۰۱۲۴۱۵	متری	۱۲۰	دوره سوم	۵۰	۴۰۰	۱۵۰۰۰۰	۵۰۰	۱۰۰۰	۱۰۰

مهر

صنایع ملامین چوب عالی کار
شماره ثبت: ۲۳۷۴۰

دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان

بانک قوامین (پلاک ۱۶۴)، طبقه پنجم، واحد ۸

تلفن: ۴۴ ۳۶ ۱۷ ۲۵ - ۴۴ ۳۶ ۱۷ ۱۴

AWI

WWW.ALIKARCO.COM

تاریخ:
شماره:
پیوست:

Required space	Ordering cost	Holding cost	Purchasing cost	Customer cost	Shortage cost	Period	Quantity	Unit	Code	Item	No.
500	3000	1000	52000	80	500	First	110	Sheet	1012816	MDF	1
500	3000	1000	52000	90	600	Second	110	Sheet	1012816	MDF	2
500	3000	1000	52000	110	800	Third	180	Sheet	1012816	MDF	3
100	300	100	30000	80	500	First	50	Kg	1020201	Adhesive	4
100	300	100	30000	90	600	Second	50	Kg	1020201	Adhesive	5
100	300	100	30000	110	800	Third	60	Kg	1020201	Adhesive	6
200	4000	500	800000	60	300	First	32	120 m	2012416	HGF	7
200	4000	500	800000	70	400	Second	32	120 m	2012416	HGF	8
200	4000	500	800000	90	600	Third	50	120 m	2012416	HGF	9
100	1000	500	150000	30	100	First	32	120 m	2012415	PVC	10
100	1000	500	150000	40	200	Second	32	120 m	2012415	PVC	11
100	1000	500	150000	60	400	Third	50	120 m	2012415	PVC	12

صنایع ملایمه چوب عالی کار
شماره ثبت: ۲۲۷۴

دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۳)، طبقه پنجم، واحد ۸
تلفن: ۳۴ ۳۶ ۱۷ ۱۴ - ۳۴ ۳۶ ۱۷ ۴۵

AWI WWW.ALIKARGO.COM

تاریخ:
شماره:
پیوست:

No. 2	No. 1	Space required	Ordering cost	Holding cost	Purchasing cost	Period	Second vendor quantity	First Vendor quantity	Unit	Code	Item	No.
500	300	250	2000	1000	52000	First	100	110	Sheet	10128-16	MDF	1
600	200	250	2000	1000	52000	Second	100	110	Sheet	10128-16	MDF	2
800	300	250	2000	1000	53000	Third	110	120	Sheet	10128-16	MDF	3
500	200	100	200	100	30000	First	70	90	Kg	1020201	Adhesive	4
600	200	100	200	100	30000	Second	40	100	Kg	1020201	Adhesive	5
800	100	100	200	100	35000	Third	110	90	Kg	1020201	Adhesive	6
300	400	300	1000	1000	800000	First	90	90	120 m	2012416	HGF	7
400	300	300	1000	1000	800000	Second	55	85	120 m	2012416	HGF	8
600	500	300	1000	1000	850000	Third	110	80	120 m	2012416	HGF	9
100	400	400	300	1000	150000	First	65	70	120 m	2012415	PVC	10
200	300	400	300	1000	150000	Second	70	70	120 m	2012415	PVC	11
400	500	400	300	1000	160000	Third	100	50	120 m	2012415	PVC	12

صنایع ملامینه چوب عالی کار
شماره ثبت: ۳۳۷۴

• دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان بانک قوامین (پلاک ۱۶۳)، طبقه پنجم، واحد ۸
• تلفن: ۴۴ ۳۶ ۱۷ ۴۵ - ۴۴ ۳۶ ۱۷ ۴۴

AWI WWW.ALIKARCO.COM



۲۴، ۵، ۹۵

« خورنار دوم »

ردیف	نام کالا	کد کالا	واحد	تعداد برای فروشده اول	تعداد برای فروشده دوم	دوره	هرینه خرید	هرینه نگهداری	هرینه سفارش	هرینه سفارش دوم	فضای انبار	کمبود ۱	کمبود ۲
۱	خام MDF	۰۱۰۱۸۱۶	ورق	۱۰۰	۱۰۰	دوره اول	۵۲۰۰۰	۱۰۰۰	۲۱۰۰	۲۰۰۰	۲۵۰	۳۰۰	۵۰۰
۲	خام MDF	۰۱۰۱۸۱۶	ورق	۱۰۰	۱۱۰	دوره دوم	۵۲۰۰۰	۱۰۰۰	۲۱۰۰	۲۰۰۰	۲۵۰	۳۰۰	۵۰۰
۳	خام MDF	۰۱۰۱۸۱۶	ورق	۱۷۰	۱۳۰	دوره سوم	۵۲۰۰۰	۱۰۰۰	۲۱۰۰	۲۰۰۰	۲۵۰	۳۰۰	۵۰۰
۴	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۵۰	۸۰	دوره اول	۳۰۰۰۰	۱۰۰	۱۸۰	۲۰۰	۱۰۰	۲۰۰	۶۰۰
۵	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۹۵	۷۰	دوره دوم	۳۰۰۰۰	۱۰۰	۱۸۰	۲۰۰	۱۰۰	۲۰۰	۸۰۰
۶	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۸۰	۶۵	دوره سوم	۳۵۰۰۰	۱۰۰	۱۸۰	۲۰۰	۱۰۰	۲۰۰	۲۰۰
۷	رول هایکلاس	۲۰۱۲۴۱۶	متری	۶۰	۶۵	دوره اول	۸۰۰۰۰۰	۱۰۰۰	۱۰۰۰	۱۱۰۰	۳۰۰	۳۰۰	۴۰۰
۸	رول هایکلاس	۲۰۱۲۴۱۶	متری	۷۰	۸۰	دوره دوم	۸۰۰۰۰۰	۱۰۰۰	۱۰۰۰	۱۱۰۰	۳۰۰	۳۰۰	۶۰۰
۹	رول هایکلاس	۲۰۱۲۴۱۶	متری	۷۰	۸۰	دوره سوم	۸۵۰۰۰۰	۱۰۰۰	۱۰۰۰	۱۱۰۰	۳۰۰	۵۰۰	۱۰۰
۱۰	رول بی وی سی	۲۰۱۲۴۱۵	متری	۶۰	۷۰	دوره اول	۱۵۰۰۰۰	۱۰۰۰	۳۲۰	۳۰۰	۴۰۰	۴۰۰	۲۰۰
۱۱	رول بی وی سی	۲۰۱۲۴۱۵	متری	۸۰	۷۵	دوره دوم	۱۵۰۰۰۰	۱۰۰۰	۳۲۰	۳۰۰	۴۰۰	۳۰۰	۲۰۰
۱۲	رول بی وی سی	۲۰۱۲۴۱۵	متری	۷۰	۹۵	دوره سوم	۱۶۰۰۰۰	۱۰۰۰	۳۲۰	۳۰۰	۴۰۰	۵۰۰	۴۰۰

سازگار

صنایع ملامینه چوب عالی کار
شماره ثبت: ۳۳۷۴۰

شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۴)، طبقه پنجم، واحد ۸
تلفن: ۴۴ ۳۶ ۱۷ ۲۵ - ۴۴ ۳۶ ۱۷ ۱۴

AWI

WWW.ALIKARCO.COM

تاریخ:
شماره:
پیوست:

No. 2	No. 1	Required space	Second vendor ordering cost	First vendor Ordering cost	Holding cost	Purchasing cost	Period	Second vendor quantity	First Vendor quantity	Unit	Code	Item	No.
500	300	250	2000	2100	1000	52000	First	100	100	Sheet	1012816	MDF	1
600	200	250	2000	2100	1000	52000	Second	110	100	Sheet	1012816	MDF	2
800	300	250	2000	2100	1000	53000	Third	130	170	Sheet	1012816	MDF	3
500	200	100	200	180	100	30000	First	80	50	Kg	1020201	Adhesive	4
600	200	100	200	180	100	35000	Second	70	95	Kg	1020201	Adhesive	5
800	100	100	200	180	100	35000	Third	65	80	Kg	1020201	Adhesive	6
300	400	300	1100	1000	1000	800000	First	65	60	120 m	2012416	HGF	7
400	300	300	1100	1000	1000	800000	Second	80	70	120 m	2012416	HGF	8
600	500	300	1100	1000	1000	850000	Third	80	70	120 m	2012416	HGF	9
100	400	400	300	320	1000	150000	First	70	60	120 m	2012415	PVC	10
200	300	400	300	320	1000	150000	Second	75	80	120 m	2012415	PVC	11
400	500	400	300	320	1000	160000	Third	95	70	120 m	2012415	PVC	12

صنایع ملایمه چوب عالی کار
شماره ثبت: ۲۲۷۴

دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۳)، طبقه پنجم، واحد ۸
تلفن: ۳۴ ۳۶ ۱۷ ۴۵ - ۳۴ ۳۶ ۱۷ ۱۴

AWI WWW.ALIKARGO.COM

۹۵/۵/۲۴

خوب سوم ۱۱

ردیف	نام کالا	کد کالا	واحد	تعداد برای فروشده اول	تعداد برای فروشده دوم	دوره	هرینه خرید	هرینه نگهداری	هرینه سفارش	هرینه سفارش	فشار ایبار	کیبود ۱	کیبود ۲
۱	خام MDF	۰۱۰۱۲۸۱۶	ورق	۱۰۰	۱۱۰	دوره اول	۵۲۰۰۰	۱۲۰۰	۳۲۰۰	۲۲۰۰	۲۵۰	۳۰۰	۵۰۰
۲	خام MDF	۰۱۰۱۲۸۱۶	ورق	۹۰	۸۰	دوره دوم	۵۲۰۰۰	۱۲۰۰	۳۲۰۰	۲۲۰۰	۲۵۰	۲۰۰	۵۰۰
۳	خام MDF	۰۱۰۱۲۸۱۶	ورق	۱۰۵	۱۰۵	دوره سوم	۵۲۰۰۰	۱۲۰۰	۳۲۰۰	۲۲۰۰	۲۵۰	۳۰۰	۵۰۰
۴	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۱۰۰	۸۰	دوره اول	۳۰۰۰۰	۱۰۰	۲۰۰	۱۰۰	۱۰۰	۲۰۰	۵۰۰
۵	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۱۰۵	۱۱۰	دوره دوم	۳۰۰۰۰	۱۰۰	۲۰۰	۱۰۰	۱۰۰	۲۰۰	۵۰۰
۶	چسب	۰۱۰۲۰۲۰۱	کیلوگرم	۹۰	۱۱۰	دوره سوم	۳۵۰۰۰	۱۰۰	۲۰۰	۱۰۰	۱۰۰	۲۰۰	۵۰۰
۷	رول هایگلاس	۲۰۱۲۴۱۶	متری ۱۲۰	۱۱۵	۱۰۰	دوره اول	۸۰۰۰۰۰	۱۱۰۰	۱۰۰۰	۱۲۰۰	۳۰۰	۴۰۰	۳۰۰
۸	رول هایگلاس	۲۰۱۲۴۱۶	متری ۱۲۰	۱۰۵	۱۰۵	دوره دوم	۸۰۰۰۰۰	۱۱۰۰	۱۰۰۰	۱۲۰۰	۳۰۰	۳۰۰	۴۰۰
۹	رول هایگلاس	۲۰۱۲۴۱۶	متری ۱۲۰	۱۰۰	۱۲۰	دوره سوم	۸۵۰۰۰۰	۱۱۰۰	۱۰۰۰	۱۲۰۰	۳۰۰	۵۰۰	۴۰۰
۱۰	رول پی وی سی	۲۰۱۲۴۱۵	متری ۱۲۰	۸۰	۸۵	دوره اول	۱۵۰۰۰۰	۱۰۰۰	۳۰۰	۳۰۰	۴۰۰	۴۰۰	۱۰۰
۱۱	رول پی وی سی	۲۰۱۲۴۱۵	متری ۱۲۰	۹۰	۹۰	دوره دوم	۱۵۰۰۰۰	۱۰۰۰	۳۰۰	۳۰۰	۴۰۰	۳۰۰	۲۰۰
۱۲	رول پی وی سی	۲۰۱۲۴۱۵	متری ۱۲۰	۸۰	۱۱۵	دوره سوم	۱۶۰۰۰۰	۱۰۰۰	۳۰۰	۳۰۰	۴۰۰	۵۰۰	۴۰۰

مدرس

صنایع چوب ملامینه عالی کار
شماره ثبت: ۲۳۷۴۰

مختار علی محمدی
شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۴)، طبقه پنجم، واحد ۸
تلفن: ۴۴ ۳۶ ۱۷ ۲۵ - ۴۴ ۳۶ ۱۷ ۱۴

AWI

WWW.ALIKARCO.COM

تاریخ:
شماره:
پیوست:

No 2	No 1	Required square	Second vendor Ordering cost	First vendor Ordering cost	Holding cost	Purchasing cost	Period	Second vendor quantity	First vendor quantity	Unit	Code	Item	No
500	300	250	2200	2200	1200	52000	First	110	100	Sheet	1012816	MDF	1
600	200	250	2000	2200	1200	52000	Second	80	90	Sheet	1012816	MDF	2
800	300	250	2000	2200	1200	53000	Third	105	105	Sheet	1012816	MDF	3
500	200	100	100	200	100	30000	First	80	100	Kg	1020201	Adhesive	4
600	100	100	100	200	100	30000	Second	110	105	Kg	1020201	Adhesive	5
800	100	100	100	200	100	35000	Third	110	90	Kg	1020201	Adhesive	6
400	300	300	1200	1000	1100	80000	First	100	115	120 m	2012416	HGF	7
300	400	300	1200	1000	1100	80000	Second	105	105	120 m	2012416	HGF	8
400	300	300	1200	1000	1100	85000	Third	120	100	120 m	2012416	HGF	9
600	500	300	1200	1000	1000	150000	First	85	80	120 m	2012415	PVC	10
100	400	400	300	300	1000	150000	Second	90	90	120 m	2012415	PVC	11
200	300	400	300	300	1000	150000	Third	115	80	120 m	2012415	PVC	12

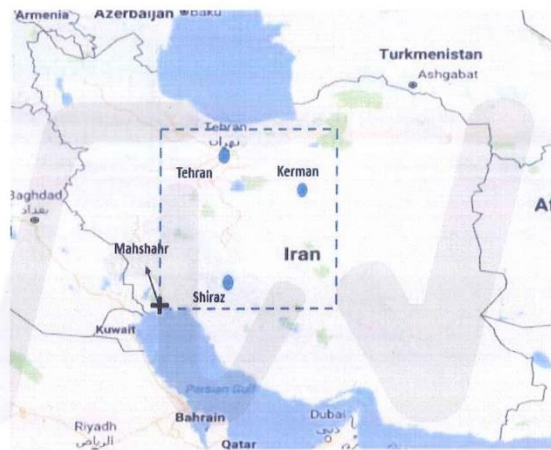
صنایع ملامینه چوب عالی کار
شماره ثبت: ۳۳۷۴

• دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۳)، طبقه پنجم، واحد A
• تلفن: ۴۴ ۳۶ ۱۷ ۱۴ - ۴۴ ۳۶ ۱۷ ۴۵

AWI WWW.ALIKARCO.COM

تاریخ:
شماره:
پیوست:

The area under study and the coordinates of the buyers' locations in Tehran, Kerman and Shiraz cities proposed by our marketing team are shown as follows:



• دفتر مرکزی: تهران، شهران، فلکه دوم، ساختمان
بانک قوامین (پلاک ۱۶۳)، طبقه پنجم، واحد ۸
• تلفن: ۴۴ ۳۶ ۱۷ ۴۵ - ۴۴ ۳۶ ۱۷ ۱۴

AWI WWW.ALIKARCO.COM

صنایع ملایمه چوب عالی کار
شماره ثبت: ۳۳۷۴

Appendix C: THE MATLAB CODES OF THE ALGORITHMS PROPOSED FOR SOLVING THE PROBLEMS

The code of GA:

```

clc
clear
close all
tic
%% Input Data:
I=input('PLZ Enter the number of buyers: ');
J=input('PLZ Enter the number of item types: ');
K=input('PLZ Enter the number of vendors: ');
T=input('PLZ Enter the number of periods: ');
% GA parameters are as following
pop=40; %population size
Gen=500; %generation number
prc=0.7; %crossover probability
prm=0.1; %mutation probability

%% Genetic Algorithm:
%Initializing
for i=1:pop
    % Input parameters of the problem are as following
    [y,a,d,h,Tt,f,s,si,C,O,Q,c,beta,mu,muper]= Input_parameters( I,J,K,T );
    % The calculations relevant to the nonlinear-integer programming including the
    objective and constraints
    [TC,b]=for_nonlinprograming(I,J,K,T,y,a,d,h,Tt,f,s,si,C,O,Q,c,beta,mu,muper);
    %Generating chromosomes
    chrom{i,1}={Q,y,TC};
    R=chrom{i,1};
    fitness(i,1)=R{3};
end
%% Selection strategy (Roulette wheel)
Best=(1,1,10^(10^100));
for iter=1:Gen
    [r1,r2]=sort(fitness); %r1 is the sorted fitness value and r2 is its rank
    W=sum(r1);
    su=0;
    Best_sol=chrom{r2(1)}; % Optimal solution
    if Best_sol{3}<Best{3}
        Best=Best_sol;
    end
end

% Tournament operator
for m=1:pop
    Re=randi(pop);
    Re1=randi(pop);
    while Re==Re1
        Re=randi(pop);
        Re1=randi(pop);
    end
    B1=chrom{Re};
    B2=chrom{Re1};
    if B1{3}>B2{3}
        sel_str{m,1}=B2;
    else sel_str{m,1}=B1;
    end
end
%% Crossover operator
e=0;
for m=1:pop
    r=rand;
    if r<=prc
        e=e+1;
        b1=randi([1,pop]); %The parent 1
        b2=randi([1,pop]); %The parent 2
        a1=sel_str{b1};
        a2=sel_str{b2};
        while a1{3}==a2{3} %Checking for a unequal parent
            b1=randi([1,pop]);
            b2=randi([1,pop]);
            a1=sel_str{b1};
            a2=sel_str{b2};
        end
    end
end

```

```

        landa=rand; %crossover for locations (y)
        x1=a1{2}*landa+a2{2}*(1-landa);
        x2=a1{2}*(1-landa)+a2{2}*landa;
        for k=1:K
            if x1(:,k)>100 %Checking for a unequal parent
                x1=100*rand;
            end
            if x2(:,k)>100
                x2=100*rand;
            end
        end

        y1=round(a1{1}*landa+a2{1}*(1-landa)); %Crossover for amount of orders (Q)
        y2=round(a1{1}*(1-landa)+a2{1}*landa);
        for i=1:I
            for j=1:J
                for k=1:K
                    for t=1:T-1
                        if y1(i,j,k,t)>150
                            y1(i,j,k,t)=randi([0,150]);
                        end
                        if y2(i,j,k,t)>150
                            y2(i,j,k,t)=randi([0,150]);
                        end
                        while y1(i,j,k,t)<(d(i,j,k,t)*(Tt(i,j,k,t)))
                            y1(i,j,k,t)=randi([0,150]);
                        end
                        while y2(i,j,k,t)<(d(i,j,k,t)*(Tt(i,j,k,t)))
                            y2(i,j,k,t)=randi([0,150]);
                        end
                    end
                end
            end
            end
            [TC1]=for_nonlinprograming(I,J,K,T,x1,a,d,h,Tt,f,s,Si,C,O,y1,c,beta,mu,muper);
            %Evaluating the offspring
            [TC2]=for_nonlinprograming(I,J,K,T,x2,a,d,h,Tt,f,s,Si,C,O,y2,c,beta,mu,muper);
            new_chrom{e,1}={y1,x1,TC1};
            new_chrom{e+1,1}={y2,x2,TC2};
            end
        end

%% Mutation operator
%%
u=0;
for m=1:pop
    w=rand;
    if w<=prm
        u=u+1;
        b3=randi(pop);
        a3=sel_str{b3};
        z1=a3{2};
        r11=randi(K);
        z1(1,r11)=rand*100; %One point mutation for locations
        z1(2,r11)=rand*100;
        z2=a3{1};
        rr1=randi(I);
        rr2=randi(J);
        rr3=randi(K);
        rr4=randi(T-1);
        z2(1:rr1,1:rr2,1:rr3,1:rr4)=randi([0,150],rr1,rr2,rr3,rr4); %One point
        mutation for amount of order (Q)

        [TC3]=for_nonlinprograming(I,J,K,T,z1,a,d,h,Tt,f,s,Si,C,O,z2,c,beta,mu,muper);
        %Evaluating the offspring
        new_chrom{e+1+u,1}={z2,z1,TC3};
        end
    end

%% Elitism operator
%%
for i=1:pop
    R22=sel_str{i,:};
    fitness1(i,1)=R22{3};
end
[h1,h2]=sort(fitness1);
for i=1:pop
    q1=sel_str{h2(i),1};
    Q1=q1{1};

```

```

        y11=q1{2};
        sel_str1{i,1}={Q1,y11,h1(i)};
    end

    bb=size(new_chrom,1);
    new_chrom1=cell(pop,1);
    for i=1:bb
        new_chrom1{i,1}=new_chrom{i,1};
    end
    %Elitism
    if bb<pop
        for i=1:(pop-bb)
            new_chrom1{bb+i,:}=sel_str1{i,:};
        end
    end
    % As a new solution
    chrom=new_chrom1;
    fitness=[];
    R=[];
    for i=1:pop
        R=chrom{i,1};
        fitness(i,1)=R{3};
    end
    B(iter,1)=Best{3};
    Y(iter,1)=iter;
    end
    %%                                End of for loop generation
    %%
    TC=Best{3}
    Q=Best{1}
    y=Best{2}

    %                                Disply Convergence Chart
    plot(Y,B)
    title('Convergence Chart' )
    xlabel('Number of Iteration')
    ylabel('Objective function Value')
    hold on
    grid on
    toc

```

The code of PSO:

```
clc
clear
close all
%% Input Data:
I=input('PLZ Enter the number of buyers: ');
J=input('PLZ Enter the number of item types: ');
K=input('PLZ Enter the number of vendors: ');
T=input('PLZ Enter the number of periods: ');
tic
% GA parameters are as following
pop=10; %population size
Gen=100; %generation number
C1=2.5; %crossover probability
C2=2.5; %mutation probability
Wmax=.9;
Wmin=.4;
delta=1;
xmin=0;
xmax=150;
% Input parameters of the problem are as following
[a,d,h,Tt,f,S,Si,C,O,c]= Input_parameters( I,J,K,T );
%% Particle Swarm Optimization Algorithm:
%Initializing
particle={};
fitness=[];
iter=[];
for i=1:pop
    [Q,y]= order( I,J,K,T,xmin,xmax,d,Tt );
    % The calculations relevant to the nonlinear-integer programming including the
    objective and constraints
    [TC]=for_nonlinprograming(I,J,K,T,y,a,d,h,Tt,f,S,Si,C,O,Q,c);
    %Generating chromosomes
    particle{i,1}={Q,y,TC};
    R=particle{i,1};
    fitness(i,1)=R{3};
end

%% Initializing velocity and position
V={};
X={};
for i=1:pop
    X(i,:)=particle(i,,:);
    V(i,1)={X{i,1}/delta};
    V(i,2)={X{i,2}/delta};
    [TC1]=for_nonlinprograming(I,J,K,T,V{i,2},a,d,h,Tt,f,S,Si,C,O,V{i,1},c);
    V(i,3)={TC1};
    fitness(i,1)=X{i,3};
end

%%      Updating Velocities and Positions

Best={1,1,10^(10^100)};
Pbest=X;
Best_sol={};
Pglo={};
r1=[];
r2=[];
for iter=1:Gen
    [r1,r2]=sort(fitness); %r1 is the sorted fitness value and r2 is its rank
    W=sum(r1);
    su=0;
    Best_sol=X(r2(1),:); % Optimal solution
    if Best_sol{3}<Best{3}
        Best=Best_sol;
    end
    Pglo=Best;

    %Updating the weight
    w=Wmax-( (Wmax-Wmin)/Gen)*iter;

    %Updating velocity
    V1={};
    for m=1:pop
        V1(m,1)={V{m,1}*w+C1*rand*(Pbest{m,1}-X{m,1})+C2*rand*(Pglo{1}-X{m,1})};
        V1(m,2)={V{m,2}*w+C1*rand*(Pbest{m,2}-X{m,2})+C2*rand*(Pglo{2}-X{m,2})};
    end

    %Updating position
```

```

X1={};
Q1=[];
y1=[];
for m=1:pop
    X1(m,1)={round(X{m,1}+V1{m,1}*delta)};
    X1(m,2)={ (X{m,2}+V1{m,2}*delta)};
    Q1=X1{m,1};
    y1=X1{m,2};
    for i=1:I
        for j=1:J
            for k=1:K
                for t=1:T-1
                    if xmax<Q1(i,j,k,t)||Q1(i,j,k,t)<xmin
                        Q1(i,j,k,t)=randi([xmin,xmax]);
                    end
                    while Q1(i,j,k,t)<(d(i,j,k,t)*Tt(i,j,k,t))
                        Q1(i,j,k,t)=randi([xmin,xmax]);
                    end
                end
            end
        end
    end
    for k=1:K
        for n=1:2
            if 100<y1(n,k)||y1(n,k)<0
                y1(n,k)=randi([0,100])*rand;
            end
        end
    end
    [TC2]=for_nonlinprograming(I,J,K,T,y1,a,d,h,Tt,f,S,Si,C,O,Q1,c);
    X1(m,1)={Q1};
    X1(m,2)={y1};
    X1(m,3)={TC2};

    % Generating Pbest
    if(X{m,3}>X1{m,3})
        Pbest(m,:)=X1(m,:);
    else Pbest(m,:)=X(m,:);
    end
end
X=X1;
V=V1;

fitness=[];
for i=1:pop
    fitness(i,1)=X{i,3};
end
B(iter,1)=Best{3};
Y(iter,1)=iter;
end
%%                                     End of for loop generation
%%
TC=Best{3}
Q=Best{1};
y=Best{2};

%                                     Disply Convergence Chart
figure(1);
plot(Y,B)
title('Convergence Chart' )
xlabel('Number of Iteration')
ylabel('Objective function Value')
hold on
grid on
figure(2);
plot(y(1,:),y(2,:),'+',a(1,:),a(2,:),'O')
xlabel('X')
ylabel('Y')
toc

```

The code of MFOA:

```

clc
clear
close all
%% Input Data:
I=input('PLZ Enter the number of buyers: ');
J=input('PLZ Enter the number of item types: ');
K=input('PLZ Enter the number of vendors: ');
T=input('PLZ Enter the number of periods: ');
tic
% GA parameters are as following
pop=10; %population size
Gen=100; %generation number
C1=2.5; %crossover probability
C2=2.5; %mutation probability
subpop=5; %The sub population of flies
Wmax=.9;
Wmin=.4;
delta=1;
xmin=0;
xmax=150;
% Input parameters of the problem are as following
[a,d,h,Tt,f,S,Si,C,O,c]= Input_parameters( I,J,K,T );
%% Particle Swarm Optimization Algorithm:
%Initializing
particle1=[];
particle={};
fitness=[];
iter=[];
for i=1:pop
    for j=1:subpop
        [Q,y]= order( I,J,K,T,xmin,xmax,d,Tt );
        % The calculations relevant to the nonlinear-integer programming including
the objective and constraints
        [TC(j,1)]=for_nonlinprograming(I,J,K,T,y,a,d,h,Tt,f,S,Si,C,O,Q,c);
        particle1{j,1}={Q,y,TC(j)};
    end
    [particle2,a1]=sort(TC)
    %Generating chromosomes
    particle(i,1)={particle1{a1(1)}};
    fitness(i,1)=particle2(1,1);
end

%% Initializing velocity and position
V={};
X={};
for i=1:pop
    X(i,:)=particle{i,,:};
    V(i,1)={X{i,1}/delta};
    V(i,2)={X{i,2}/delta};
    [TC1]=for_nonlinprograming(I,J,K,T,V{i,2},a,d,h,Tt,f,S,Si,C,O,V{i,1},c);
    V(i,3)={TC1};
    fitness(i,1)=X{i,3};
end

%%          Updating Velocities and Positions

Best={1,1,10^(10^100)};
Pbest=X;
Best_sol={};
Pglo={};
r1=[];
r2=[];
for iter=1:Gen
    [r1,r2]=sort(fitness); %r1 is the sorted fitness value and r2 is its rank
    W=sum(r1);
    su=0;
    Best_sol=X(r2(1),:); % Optimal solution
    if Best_sol{3}<Best{3}
        Best=Best_sol;
    end
    Pglo=Best;

    %Updating the weight
    w=Wmax-(Wmax-Wmin)/Gen*iter;

```

```

%Updating velocity
V1=[];
for m=1:pop
    V1(m,1)={V{m,1}*w+C1*rand*(Pbest{m,1}-X{m,1})+C2*rand*(Pglo{1}-X{m,1})};
    V1(m,2)={V{m,2}*w+C1*rand*(Pbest{m,2}-X{m,2})+C2*rand*(Pglo{2}-X{m,2})};
end

%Updating position
X1=[];
Q1=[];
y1=[];
for m=1:pop
    X1(m,1)={round(X{m,1}+V1{m,1}*delta)};
    X1(m,2)={ (X{m,2}+V1{m,2}*delta)};
    Q1=X1{m,1};
    y1=X1{m,2};
    for i=1:I
        for j=1:J
            for k=1:K
                for t=1:T-1
                    if xmax<Q1(i,j,k,t)||Q1(i,j,k,t)<xmin
                        Q1(i,j,k,t)=randi([xmin,xmax]);
                    end
                    while Q1(i,j,k,t)<(d(i,j,k,t)*Tt(i,j,k,t))
                        Q1(i,j,k,t)=randi([xmin,xmax]);
                    end
                end
            end
        end
    end
    for k=1:K
        for n=1:2
            if 100<y1(n,k)||y1(n,k)<0
                y1(n,k)=randi([0,100])*rand;
            end
        end
    end
    [TC2]=for_nonlinprograming(I,J,K,T,y1,a,d,h,Tt,f,S,Si,C,O,Q1,c);
    X1(m,1)={Q1};
    X1(m,2)={y1};
    X1(m,3)={TC2};

    % Generating Pbest
    if(X{m,3}>X1{m,3})
        Pbest(m,:)=X1(m,:);
    else Pbest(m,:)=X(m,:);
    end
end
X=X1;
V=V1;

fitness=[];
for i=1:pop
    fitness(i,1)=X{i,3};
end
B(iter,1)=Best{3};
Y(iter,1)=iter;
end
%%                                End of for loop generation
%%
TC=Best{3}
Q=Best{1};
y=Best{2};

%                                Disply Convergence Chart
figure(1);
plot(Y,B)
title('Convergence Chart' )
xlabel('Number of Iteration')
ylabel('Objective function Value')
hold on
grid on
figure(2);
plot(y(1,:),y(2,:),'+',a(1,:),a(2,:),'O')
xlabel('X')
ylabel('Y')
toc

```